中华学习机
使用技巧与检修360问

沈大林　台飚　董乐　艾伦　编



電子工業出版社

中华学习机
使用技巧与检修360问

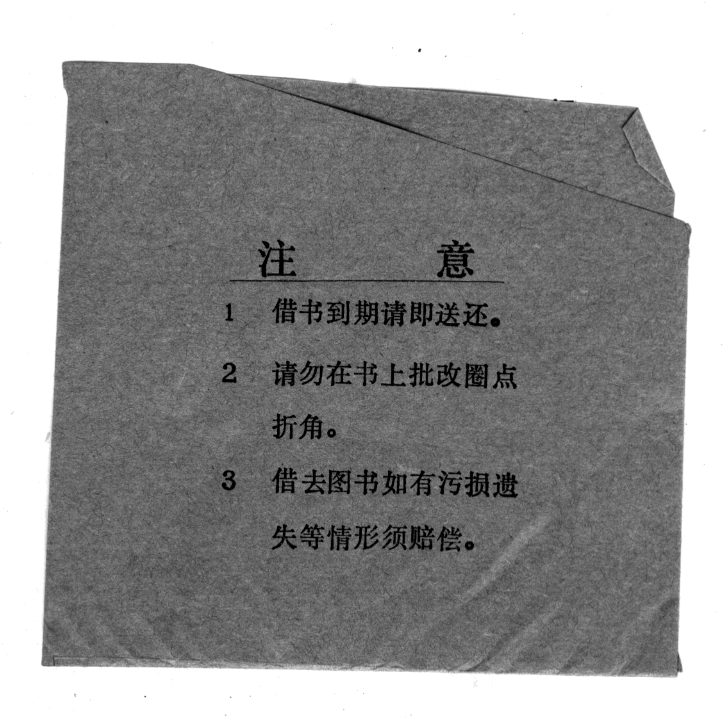
沈大林　台飚　董乐　艾伦　编

电子工业出版社

内 容 提 要

本书以问题的形式对中华学习机进行了全面的介绍，内容涉及选购、安装、调试，中华学习机 BASIC 语言与汇编语言、LOGO 语言、PASCAL 语言、dBASEⅡ的使用及其编程技巧，磁盘操作系统、CP／M 操作系统、中华学习机的汉字系统，中华学习机图形显示系统，打印机的使用，磁盘文件的加密、解密与复制，Z80／PRT 卡、万能接口卡的使用，中华学习机主板、电源、驱动器、打印卡的故障检修等各个方面。

读者对象：青少年、广大的中华学习机用户



前 言

目前，世界上许多先进国家计算机教育的重点已从高等院校转向中小学校和家庭，计算机已逐步形成普及的趋势。邓小平同志曾指出："计算机的普及要从娃娃抓起"，这一指示具有伟大的战略意义，它促进了我国计算机教育的发展，有利于我国科学技术的发展和经济的腾飞。中华学习机正是在这种形势下，由我国自行研制开发和生产的。中华学习机适用于中小学校和家庭，是广大青少年学习和娱乐的好伙伴。

CEC-I 中华学习机与 APPLE IIe 微机兼容，其功能比 APPLE IIe 微机有所增强。它可以运行 APPLE II 微机的各种软件，还可以享用 APPLE II 微机的大量接口卡。它增强的功能有：键盘字符增多；可以识别小写字母；有 64K 内存贮器，不需再加 16K 扩展卡；可用键盘操作进行主机自检诊断；可配置光标定位器；具有汉字处理功能；可进行双倍高分辨图形显示和 80 行文本显示；恢复单步跟踪命令，加入小汇编功能；增加了 BASIC 语言的功能等。

CEC-I 中华学习机具有汉字系统，提供拼音和区位两种汉字输入方法，主机内存配有全点阵汉字字库，提供一、二级汉字字库，具有 6763 个汉字。屏幕可显示 10 行汉字，每行 17 个汉字或 34 个 ASCII 码字符，全屏幕可显示 170 个汉字。汉字字型采用 16 * 16 点阵，在打印机上可打印 16 种汉字字型。

CEC-I 中华学习机主机上固化有监控程序，BASIC 语言解释程序，中文 BASIC 语言解释程序和 LOGO 语言解释程序。开机后，用户就可以方便地使用这些语言，而不需要从外部设备上将各种程序读入主机内存中。CEC-I 中华学习机的结构配置灵活，价格低廉。它的主机配上家用彩色或黑白电视机及盒式录音机就可组成基本的系统使用，如果将它的主机与彩色电视机或监视器、软盘驱动器、打印机配合使用，可以构成一台功能较强，而且具有汉字支持的微机系统。

最近，又推出了新型号的中华学习机 CEC-I PLUS。该机器内存量扩大为 128KB RAM，128KB ROM，ROM 中存有监控程序、CEC-BASIC 解释程序、汉字管理系统和扩充的汉字输入方式等。CEC-I PLUS 微机的汉字功能加强了，它有国标二级汉字字库，具有拼音、区位码和五笔字型等多种汉字输入方式。并可以方便地配接其它汉字输入方式。

《中华学习机使用技巧与检修 360 问》一书由"中华学习机概述"、"中华学习机 BASIC 语言与汇编语言"、"打印机的使用"、"中华学习机图形显示"、"磁盘操作系统"、"磁盘加密、解密与复制"、"汉字操作系统"、"CP / M 操作系统"、"dBASE II"、"APPLE-PASCAL"、"LOGO 语言"和"硬件"等十个部分三百六十个问题组成。涉及中华学习机的基本知识、基本使用方法、各种语言和操作系统的使用技巧、部分外设使用、硬件结构、维修技

巧、接口制作等内容。它能指导你怎样安装与使用中华学习机，指导你了解各种语言和操作系统的使用技巧，指导你对程序和磁盘文件进行加密解密，指导你掌握各种绘图技巧，指导你对中华学习机主机、驱动器、显示器和打印机进行维修，指导你制作维修工具和接口卡等。

《中华学习机使用技巧与检修360问》一书采用问答的形式，简单明了地讲述了360个具体的使用技巧。它既能帮助初学者迅速掌握中华学习机的基本使用方法和基本常识，又能帮助已具有一定水平的使用者进一步提高水平和掌握更多的使用方法与技巧。不管是初学者还是已具有一定水平的使用者，都将从本书中获得新的知识，吸取新的内容与经验。

本书内容通俗易懂，具有很强的实用性与趣味性。其中"中华学习机概述"、"中华学习机BASIC语言与汇编语言"、"打印机的使用"和"中华学习机图形显示"四部分由沈大林同志编写，"磁盘操作系统"、"磁盘加密、解密与复制"、"汉字操作系统"三部分由台飚同志编写，"CP／M系统部分"、"dBASE Ⅱ部分"、"APPLE-PASCAL部分"由董乐同志编写，"硬件部分"由艾伦同志编写。在编写当中，还得到北京136中电子技术职业高中老师和学生，李娅、吕旭东、许睿、宋锦、李玉玲、韦英魁、徐玉梅的大力帮助，在此表示感谢。

作者衷心希望这本书能成为每一个中华学习机使用者的学习参考书，能为每一个中华学习机使用者提供点滴启发与帮助。也衷心地希望每一位读者能给本书提出宝贵的意见与建议。

编　者

1990年6月

目　　录

一、中华学习机概述

1. 中华学习机有哪些特点

中华学习机是由电子工业部计算机与信息局组织，清华大学主持联合设计的八位普及型微型计算机，CEC 型中华学习机主要是面向中小学、面向家庭、面向成人教育与幼儿智力开发。

CEC 型中华学习机主要具有以下特点：

1. 与 APPLE Ⅱe 微机兼容

APPLE Ⅱ 微机于 1975 年由美国加利福尼亚州旧金山市的约伯·史蒂芬和奥斯尼雅克·史蒂芬两兄弟研制成功的，1977 年正式投入生产，APPLE Ⅱe 是 1983 年投入生产的，它是 APPLE Ⅱ 微机的增强型。十几年来 APPLE Ⅱ 微机遍布全世界，且经久不衰，其原因是：它系统设计成功，硬件具有积木式的扩展功能。具有大量的外围扩展卡和成千上万的应用软件，具有较好的图形功能，而且功能不断增强，应用范围不断扩展。

APPLE Ⅱe 微机是在 APPLE Ⅱ PLUS 微机基础上发展起来的，它的集成度高了，功能增强了，增强的功能主要有：可以识别小写字母；键盘字符增多；按字符键的时间超过一秒钟后能连续自动重复执行该键功能；有 64K 内存贮器，不需再加 16K 内扩展卡；增加 80 行文本显示和双倍高分辨率图形显示；可用按键操作进行主机自检诊断；可配置光标定位器等。

CEC—Ⅰ型中华学习机是在 APPLE Ⅱe 微机基础上设计开发的新机种，它与 APPLE Ⅱe 兼容，可共享已有的大量外围扩展卡的硬件资源与丰富的应用软件资源。

2. 增加汉字处理功能

中华学习机具有汉字系统，提供拼音和区位两种汉字输入方法，主机内存配有全点阵汉字字库，提供一、二级汉字点阵，具有 6763 个汉字。屏幕可显示 10 行汉字，每行 17 个汉字或 34 个 ASCII 字符，满屏可显示 170 个汉字。汉字字型采用 16×16 点阵。

汉字系统提供 15 种字型供用户打印时使用，常用的有八种，同时允许用户设置行距、字距和每行的字数。另外还建立了中文 BASIC 语言，可以处理汉字。

3. 增加自检功能，扩充编辑功能键

CEC 中华学习机增加了对主机的自检功能，这样有利于生产和检修，方便了生产厂家和用户。同时，为了使用方便，增加了一些 APPLE Ⅱ 微机没有的功能，例如 ↑ 和 ↓ 键。

4. 增加磁盘假读功能

在 CEC－Ⅰ型中华学习机监控系统中，加入了一段假读软盘命令，以判别系统是否接有软盘驱动器和是否将磁盘装入磁盘驱动器中。当上述判别条件中有一个不满足时，监控便不会转入软盘驱动器，而直接进入 BASIC 状态。这样可避免没装入磁盘情况下启动主

机出现的死循环现象。

CEC－Ⅱ型中华学习机还可以将固化有磁盘驱动程序的 ROM 集成块插在主机板上，此时系统不一定接有磁盘驱动器。

5. 恢复单步跟踪命令，加入小汇编功能

在 APPLE Ⅱ 监控的最初版本中，有单步命令 S 和跟踪命令 T 可供用户调试程序时使用，但在后来的自启动监控的版本中，由于空间的原因而删去了这两个功能，在 CEC－Ⅰ中华学习机中恢复了这两个有用的监控命令。另外，小汇编程序是用户经常使用的一个软件，CEC－Ⅰ型中华学习机也将其固化在 ROM 中，可以由监控直接进入小汇编状态。

6. 增加了 BASIC 语言的功能

中华学习机的 BASIC 语言（CEC－BASIC）固化在主机电路板的 ROM 中，它除了具有 APPLESOFT BASIC 语言的功能外还具有汉字处理功能。另外，还增添了三条语句，扩展了存取磁带程序、执行游戏程序和发音方面的功能：

(1) 用户可利用家用收录机对磁带进行存取程序。原 APPLESOFT BASIC 语言对磁带的存取虽然提供了 LOAD 和 SAVE 两条命令，但存入磁带的程序不能命名，取出来时也不能按名字选取需要的程序。为此，CEC－BASIC 中加入了按文件名存取磁带的功能。这样，用户可敲入 SAVE“文件名”将程序以确定的名字存入磁带中；也可敲入 LOAD“文件名”将磁带中“文件名”所表示的程序自动取出来。使操作简单、方便。

(2) 将游戏程序转录到磁带后，没有配备磁盘驱动器的用户便可以使用这些游戏。用户只需敲入“PLAY”这一语句便可将游戏程序装入内存并立即开始运行。

(3) 为了方便编音乐程序的用户，在 CEC－BASIC 语句中提供了一条新语句 MUSIC。只要给该语句以合适的参数，就可以使计算机奏乐。

7. 增加 LOGO 语言，具有多种接口

CEC－Ⅰ型中华学习机将 LOGO 解释程序固化在主机 ROM 中，用户在开机后键入 LG，即可使机器自动进入 LOGO 状态。

CEC－Ⅰ中华学习机具有单色监视器视频输出接口，PAL 制式黑白和彩色电视机射频接口，与 APPLE Ⅱe 兼容的游戏棒和磁带录音机接口，以及在主机板上保留了一个 50 芯 I／O 扩充插口和一个驱动器接口。

8. 整机设计合理、集成度高和性能价格比高

2. 中华学习机系统由哪几种设备组成

一套完整的中华学习机系统由主机（包括键盘）、荧光屏显示器、磁带机（或磁盘机）与打印机组成。

1. 主机与键盘（Keyboard）

主机内部有中华学习机主线路板，它主要由主控部分、汉字系统、磁盘驱动器接口、PAL 制形成电路和键盘接口等电路组成。在主机壳内左边有一个开关电源，输入电压为交流 220V，最大功率不大于 25W。主机外壳上还装有键盘。

键盘是一种输入设备，具备标准打字机键盘的各种键，共 69 个键，包括大小写字母、数字和一些功能键，可用手击键向计算机输入信息。

2. 荧光屏显示器（Cathode Rey Tube）

这是电脑的输入设备，简称为CRT。用它可以显示计算机的有关信息，例如用户从键盘上敲入并为主机所接收的字符、程序、电脑向用户的提示、程序的运行结果等。

荧光屏显示器有专为电脑设置的黑白、绿色或彩色显示器，也可以是一般家用黑白或彩色电视机。中华学习机屏幕显示有三种方式，即文本显示、低分辨率图形显示和高分辨率图形显示。文本显示就是字符显示，每幅屏幕可显示24行、每行40个字符，共960个字符。每个字符由5X7个象素点组成，可显示64个字符。每个字符又有正常（Normal即黑底白字）、反转（Inverse即白底黑字）和闪烁（Flash即一会黑底白字，一会白底黑字）三种显示方式。低分辨率图形显示可以把屏幕分为48行×40列＝1920个色块，每个色块可以有16种颜色。高分辨率图形是将屏幕分为192行×280列＝53760个象素点，每个象素点可有六种颜色。

3. 磁带机（Cassette Tape Record）

磁带机就是通常的卡式录音机。用它可以存贮程序和数据，存取信息的过程与录放音乐和声音的过程类似。采用磁带机可使成本下降，但存取信息慢，查找文件困难，使用不方便。

4. 磁盘机（Disk Drive）

磁盘机也叫磁盘驱动器。用它来驱动磁盘，可将信息存入磁盘及从磁盘中读取信息。磁盘机的价格远高于磁带机的价格，但它存取信息的速度高、存贮量大、使用方便、查找文件容易、存贮信息可靠性高。

5. 打印机（Printer）

打印机也叫行印机或列表机。它主要有三大类型：活字击打式、针式和非机械式（例如喷墨式和热灼式等）。目前主要采用针式打印机，例如MX−80、MX−100、FX−80、FX−100和CP−80等。在针式打印机中，有的以打印字符为主，它的针数较少，速度较高；有的以打印汉字为主，针数较多（如24针），速度较慢。MX与FX系列打印机均是EPSON公司的产品，后者是在前者的基础上发展起来的，速度较快，性能较好。FX−80与FX−100的主要区别是打印宽度不同，FX−80打印机一行可打印80个字符，而FX−100打印机一行可打印136个字符。

3. 使用中华学习机应注意哪些事项

中华学习机的寿命和损坏率与使用环境和使用方式有着直接的关系。因此，在使用它时应注意以下几点：

1. 对使用环境应有一定的要求

虽然中华学习机是普及型微电脑，但由于它的电子元件大多是采用TTL或MOS电路，所以对使用环境有一定的要求。一般环境温度应在14°c～30°c之间，最好的环境温度为22°c±2°c。当温度过高时，机内散热条件差，温升快，会使主机工作不稳定，使驱动器工作不正常；当温度过低时，驱动器对磁盘读写信息易产生错误。一般环境相对湿度为40%～75%之间，当相对湿度过高时，机器内部元件易受潮损坏、电子器件内部杂散电容增大、漏电电流加大；当相对湿度过低时，机器内易产生静电积聚，击穿MOS

电路，使机器工作不正常或损坏。

另外，中华学习机还要求有一个尘埃少的干净环境。尘埃通常带有酸性，一旦进入机内，和空气中的水分相互作用，会腐蚀机内元件，在环境相对湿度较高时。尘埃也带有静电荷，一旦进入机内会对内部元件进行破坏，尤其是易损坏 MOS 电路。尘埃如果沾附在磁盘上或进入驱动器，会使驱动器读写磁盘信息产生错误。

除以上条件外，中华学习机还应避开腐蚀性强的气体和强磁场、强电场。

2. 对交流电源的要求

供给中华学习机的交流电源不应过高或过低，一般应在 210V～230V 之间。电压过高，会使电源调整功率加大，增大机内热量，使机器工作不正常，严重时会损坏机内稳压电源和其它器件；电压过低，会使机内稳压电源电流增大，使机器工作不正常，使显示器工作不稳定，容易使驱动器读写磁盘信息产生错误。

供给中华学习机的交流电源还应该稳定，如果电压忽高忽低，会使整个计算机系统工作不正常，甚至造成机器的损坏。

为此，中华学习机的交流电源不应与动力交流电源接在一起，最好采用交流稳压器供电。由于中华学习机主机功耗 25W，显示器功耗约 50W。小系统总功耗约 75W，所以在使用 1～13 台中华学习机时可以采用 1000VA 较小的交流稳压器供电。

3. 使用主机的注意事项

(1) 主机接通电源后，不要打开机箱盖板抚摸机内集成电路片，以免人体感应的电荷对 MOS 电路造成损坏；也不要带电、插接口卡，以免使主机和接口卡损坏。

(2) 不要用过大的力量敲击键盘，使键盘损坏。

(3) 主机不用时，应盖上防尘罩，以免尘埃进入机内。

(4) 定期对主机进行清尘处理和定期进行测试试验，及时发现问题及时解决。

4. 使用显示器的注意事项

使用显示器时，除了注意交流电源电压应正确和不要频繁按动电源开关外，还应特别注意显示屏的亮度不要太大，这不但会使人产生疲劳，而且会加速显示管的衰老，缩短使用寿命。

5. 使用打印机的注意事项

(1) 打印机应放置平稳，所供交流电源应稳定。

(2) 打印纸和色带安装应正确，打印纸应能活动自如（这可以通过调节手轮来检验），打印色带应在打印机工作时缓慢移动，（这可以通过转动色带左边的旋转手柄来检验），若色带不移动，会使打印头总在色带的一段上打印，造成色带损坏。

(3) 打印头在横杆上滑动应自如，否则会使打印头工作时发热，缩短使用寿命。为此，应使横杆保持清洁，定期擦拭。

6. 使用磁盘驱动器的注意事项

磁盘驱动器结构精密、价格昂贵、维修困难，使用时应特别小心注意。

(1) 磁盘驱动器应轻拿轻放，避免剧烈震动。使用时应安放平稳。运输或长期不使用时，应在驱动器内插入硬纸盘，关闭驱动器的门，以免尘埃进入驱动器。

(2) 磁盘驱动器工作时，不要移动。

(3) 不要在马达旋转时（驱动器指示灯亮时），插入或拿出磁盘

(4) 磁盘驱动器应有较好的工作环境，尤其要求尘埃少，远离强磁场。为了减少驱动器内的尘埃，要求不要将沾有尘埃的磁盘放入驱动器内。要定期用清洁磁盘清洗磁头。

7. 使用软磁盘的注意事项

软磁盘是用来存贮程序、数据和文件等信息的介质，因为它易损坏，所以使用时应特别小心。

(1) 绝对禁止用手触摸磁盘读写窗口（也叫存取洞）内磁盘的暴露部分，也应防止尘埃沾污。

(2) 平时取磁盘应拿它的边沿，不要加重压，不可弯曲。不用时应及时插入纸外套中，并立放在盒中，以免对磁盘表面施加压力。

(3) 标签应写好后再贴到磁盘上，不要贴上标签后，再在磁盘上写字，以免损坏磁盘。

(4) 在磁盘驱动器指示灯亮时，绝对不能将磁盘从驱动器中取出来，或将磁盘放入驱动器中。

(5) 磁盘不用时，应将它从磁盘驱动器中取出来。

(6) 应让磁盘远离扬声器、变压器和马达等产生的磁场，远离热源，避免阳光直射。不用的磁盘应存放在阴凉干燥的地方。

4. 如何安装中华学习机

安装中华学习机系统前，应关闭主机和各外部设备的电源（将电源开关置于 OFF 位置），检查主机和各外部设备有无明显的损坏，将磁盘驱动器内保护磁头的硬纸片取出，然后按下述步骤进行安装：

第一步：将监视器（彩色、墨绿或黑白）或电视机（彩色或黑白）与主机相连。

如果是监视器，可将视频电缆线一头插入主机“监视器”接口，一头插入监视器 IN 插座。如果电视机，可按下述次序连接：

(1) 将电视机天线插头拔掉。

(2) 将视频电缆线一头插入主机“电视机”接口，一头插入电视机天线插座，天线插座的输入阻抗为 75Ω。

(3) 将电视机音量调至最小处。

第二步：将录音机与主机相连

把录音机电缆线的一端（五芯插头）插入主机侧面的录音机接口，应注意五芯插头的凹口部分在正上方。电缆线另一端的两个 3.5mm 插头，分别插入录音机 EAR 插孔（连接主机 IN 插孔）和 MIC 插孔 （连接主机 OUT)。

第三步：将驱动器与主机相连

把磁盘驱动器电缆线（20 线扁平电缆）一端插座连接驱动器，另一端连接主机上的驱动器接口。必须注意：插针与插孔必须一一对位，电缆线两端的凸出部分都朝上。

第四步：将接口卡插入主机扩充槽中

主机内右上方有一个 50 线的扩充槽，可插入各种接口卡（如打印卡，A／D 转换卡等）。接口卡尺寸太大插不进去时，可采用 50 线转接板把槽口抬高，再插接口卡。

将接口卡按元件面朝前对准槽口往下插。必须注意：一定要对位垂直插入。插入接口卡后还应根据需要选定扩充槽槽号，槽号可以为1,2,4,5,7中任一个，用户可以用短接幅来选择。3号槽已定于汉字系统，6号槽定于驱动器接口，所以不可选用。出厂时一般都设在第一槽，即将两个短接符插在标有“1”的位置上。选择槽号插头位于主机板的右侧，并标有槽号。如需改槽号，需打开主机壳，把原在一槽位置的两个短接帽取下并插入你所需要的槽号插头上。

第五步：将游戏棒与主机相连

九芯游戏棒插座位于主机右侧，把游戏棒插头座直接对准游戏棒接口插入。插入时要注意插针与插孔一一对应。

第六步：接通电源校准电视机频道

将主机、显示器、录音机电源插头插入电源插座中，并将主机、显示器电源开关拨至“ON”处。一般应先开显示器再开主机。

如果使用的是黑白电视机，则按厂家规定的电视频道，调准电视机的频道。再调节频道微调及亮度、对比度旋钮，直到屏幕出现清晰的“ZHONG HUA XUE XI JI”字样为止。如果使用的是彩色电视机，同样先选好频道，再通过自检方式进入彩条图象（参看问题5），然后调节频道微调及饱合度、亮度、对比度旋钮，直至屏幕上的彩条满意为止。以后可把这频道定为计算机专用频道，不需再调整了。

5. 如何启动中华学习机和进行机器自检

1. 基本系统的启动

所谓基本系统就是只有主机与显示器的系统。

首先按要求将主机与显示器连接好，再打开主机与显示器的电源。开机时可听到“哗”的一声，这声音告诉用户，主机已准备好。同时键盘右上方指示灯发亮。如果一切正常，主机应自动进入西文CEC−BASIC语言状态，屏幕有一闪烁的小方块，方框左边有CEC−BASIC语言提示符“]”。闪烁的小方块叫光标，它指示出键盘输入字符的显示位置。在光标与提示符上方有一串字样“ZHONG HUA XUE XI JI VERSION 1.1”。

2. 连接驱动器的系统启动

首先按正确的方法连接好系统。将DOS3.3系统盘水平缓慢插入磁盘驱动器中（磁盘缺口应朝左，并在磁盘完全插入后关上磁盘机小门。然后打开主机、显示器电源。显示器应首先显示：

```
ZHOHG HUA XUE XI JI
VERSION 1.1
]
```

同时，磁盘驱动器上指示灯亮，并可以听到马达转动的声音，直到指示灯熄灭，屏幕显示如下：

```
DOS VERSION 3.3           08/25/80
APPLE II PLUS OR ROMCARD SYSTEM MASTER
(LOADING INTEGER INTO LANGUAGE CARD)
]
```

此时中华学习机已处于西文 BASIC 状态下。如果没插入磁盘系统会直接进入西文 BASIC 状态。这时如需装入 DOS 系统，可将 DOS 盘插入驱动器中，再键入：PR#6，等一会后，屏幕会显示与上面相同的字幕。

3. 中西文切换与系统复位

(1) 中西文切换

接通电源后，机器处于西文显示状态。如果键入"中文"键，机器就进入到汉字显示状态。屏幕可显示 10 行。每行 17 个汉字。在屏幕底部的第 11 行为状态行。屏幕正中出现"中华学习机 CEC-Ⅰ"字样，在状态行左边显示"字母:"字样，表示系统处于字符输入状态，可输入 BASIC 程序和各种字符。当敲入 F2 时，状态行左端显示"拼音:"，系统处于汉字拼音输入方式；当敲入 F3 时，状态行左端显示"区位:"，系统处于汉字区位输入方式。如果按下 F1 键，系统又回到字符输入状态。按 CTRL-O 键可以控制状态字符的显示与不显示，按一次 CTRL-O 可使状态字符消失，再按一次 CTRL-O 又使状态字符恢复。

在中文方式下，敲入"西文"键，机器就退出中文显示方式，进入西文显示方式。除通过键盘使机器进行中西文切换外，还可以在 BASIC 程序中进行这种切换。在 BASIC 程序中使用 PR#3 命令可使机器由西文状态进入中文状态，使用 TEXT 或 PRINT CHR $ (17) 即可退出中文状态进入西文状态。

如果因操作失误或程序错误等原因造成系统进入死循环时，可同时按下 CTRL-RESET 键，使系统重新启动 CEC-BASIC。这一过程叫系统复位。

4. LOGO 语言的进入

使主机由 BASIC 状态进入 LOGO 语言的方式是：

(1) 在系统未装有 DOS 的 BASIC 状态下，键入：

```
] LG
```

即可进入 LOGO 语言状态。这时屏幕显示：

```
GHINESE  EDUCATION  COMPUTER
        LOGO
        VERSION 1.1
        1987.6
?
```

其中"?"号是 LOGO 语言的提示符。

(2) 在系统已装有 DOS 的 BASIC 状态下，先键入：

```
] MAXFILESI
```

再键入:

] LG

即可进入 LOGO 语言状态。

5. 主机的自检

同时键入 CTRL−RESET−TEST 三键，最后释放 TEST 键，可以进行主机自检，将对 RAM 及 ROM 进行检测。

显示器显示相应的检测内容和结果:

```
                 MEMORY−TEST
                            TIMES: 000
RAM         BFFF          OK
BNK1        FFFF          OK
BNK2        FFFF          OK
ROM1        BFFF          LOGO
ROM2        FFFF          CEC−BASIC
AUX1        BFFF          HZTABLR
AUX2        FFFF          HZPROGRAM
AUX3        5D99          CECW1
```

其中 RAM 部分是测 $0000～$BFFF 48K RAM。BNK1、BNK2 部分是用两个软开关测试 $D000～$FFFF RAM。当测试正确时，屏幕显示“OK”；如果有错，则屏幕相应处显示“ERR”

ROM1 是测试 LOGO 语言的解释程序，ROM2 是测 CEC−BASIC 解释程序和监控内容的。

AUX1 测试汉字码表内容，AUX2 测试汉字管理系统内容，AUX3 测试两块 1 兆位的汉字库内容。

如果测试正确则在相应位置显示测试内容的名字，如果测试不正确，则屏幕显示“UNKNOWN”字样。当没有加汉字库时，屏幕显示“NULL”字样。

当 ROM、RAM 测试完毕后，程序进行彩色显示测试，屏幕将显示 16 种不同颜色的彩条。自检程序可连续循环地测试，并在 TIME 右边显示测试次数。如果需要调试彩电，可在屏幕显示彩条时按任何键，使屏幕一直显示彩条，直到调试完毕为止。调试完毕后，再按下 CTRL−RESET 键，机器即进入 CEC−BASIC 西文状态。

6. 如何使用中华学习机键盘按键

中华学习机的键盘有 69 个按键，其中有 48 个字符键，有 21 个功能键。它们的使用如下:

1. CAPSLOCK 键

当按下此锁定键后，26 个英文字母为大写；当抬起该键后，所有再按的英文字母都是小写。

2. SHIFT 键和字符键

键盘上一些字符键有上、下两个字符。当只按字符键时，键盘输入的是下面的字符。当按下 SHIFT 键后再按字符键，键盘输入的是上面的字符。例如：按下 SHIFT 键后再按！与 1 所在的键，则键盘输入的是上面的字符！；如果只按该字符键，则键盘输入的是字符 1。

3. CTRL 键

CTRL 键为控制键，它与其它键同时按下，会形成某种控制功能。例如：

(1) CTRL−S：可使屏幕滚动显示的程序或运行结果暂时停止，以便使人们看清屏幕显示的内容。如再按一次，暂停结束，程序继续显示或运行结果继续显示。

(2) CTRL−B：使计算机由监控状态回到进入监控状态前的 BASIC 语言状态，并破坏当前的 BASIC 程序和变量。

(3) CTRL−C：在 BASIC 状态下可中止程序的运行和列程序。中止程序运行时好象执行了一条 STOP 语句，使用 CONT 命令可使程序继续运行（在整数 BASIC 语言中使用 CON 命令）。在监控状态下，使用它可以在不破坏当前 BASIC 程序和变量的情况下返回到 BASIC 语言状态。

(4) CTRL−G：使扬声器发出一声"嘟"。

(5) CTRL−H：光标左移一格，相当于按一下←键。

(6) CTRL−U：光标右移一格，相当于按一下→键。

(7) CTRL−M：相当于按下 RETURN 回车键。

(8) CTRL−J：使光标移至下一行首位。

(9) CTRL−X：使刚键入的一程序行的字符无效。

以上各种控制除了 CTRL−B 外，均可用于 BASIC 和监控状态。

4. ESC 键

ESC 本是 ESCAPE（擦去）的缩写。当按下 ESC 键，再按下 SHIFT 和@与 2 所在的键时，即可以将屏幕显示的字符清除干净。ESC 键除了有擦去功能外，更重要的是它和另外 8 个键配合而实现编辑作用，以及和另外 3 个键配合实现擦去功能。它与 CTRL 用法不同，应先按 ESC 再按其它键，而 CTRL 键是与其它键同时按下，同时放开。

(1) ESC−A：使光标右移一位，退出编辑状态。

(2) ESC−B：使光标左移一位，退出编辑状态。

(3) ESC−C：使光标下移一行，退出编辑状态。

(4) ESC−D：使光标上移一行，退出编辑状态。

(5) ESC−E：将光标处至本行尾的所有字符清除。

(6) ESC−F：将光标处至屏幕末尾的所有字符清除。

(7) ESC−@：将整个屏幕的字符都清除。

(8) ESC−I：使光标上移一行，保持编辑状态。

(9) ESC−J：使光标左移一位，保持编辑状态。

(10) ESC−K：使光标右移一位，保持编辑状态。

(11) ESC–M：使光标下移一行，保持编辑状态。

使用前四个编辑命令的特点是：在按下 ESC 键后，机器进入编辑状态，按一下 A、B、C、D 中某个键后，光标会作相应的移动，同时使计算机脱离编辑状态。如果再按 A、B、C、D 中某个键，屏幕光标移动不再按原规律进行，而是在屏幕上显示与所按键名一致的字符。

使用后四个命令有这样的特点：在按下 ESC 键后，机器进入编辑状态，这时按 I、J、K、M 中任何一个键都可使光标作相应的移动，只有在键入其它键后才会使机器退出编辑状态。利用上述后四个命令可以很方便地对程序进行修改。

修改程序，可先按 ESC 键使机器进入编辑状态，再按 I、J、K、M 键使光标移至要修改的程序行的起始位置，然后退出编辑状态，用→键使光标移至要修改处进行修改。→键使光标移动，经过的字符均送入键盘缓冲区中，就好象刚刚敲入这些字符一样。而编辑状态下，光标经过的字符没有送入键盘缓冲区。

5. TEST 键

同时按下 CTRL–RESET–TEST 三键，可使主机进行自检。

6. RESET 键

它与 CTRL 键同时按下时，可使系统复位。

7. RETURN 键

按该键可使光标移至下一行首位，可结束一个命令，将命令送入计算机让计算机执行；也可结束一个程序行的程序，将程序由键盘缓冲区送入计算机内的程序区中。一般把它叫回车键，可用“↓”表示。

8. QUIT 键

按此键可使计算机从监控状态反回 BASIC 状态，在此过程中不破坏内存里原有的 BASIC 程序。它相当于按下 CTRL–C 键。

9. TAB 键

它象打印机中的定位键，例如屏幕上定了 1、11、21、31 四个位置，光标在 13 处，则按 TAB 键后会使光标移至定位 21 处。

10. 其它键

(1) F1：进入字符输入方式。

(2) F2：选定汉字输入的拼音方式。

(3) F3：选定汉字输入的区位方式。

(4) F4：供用户使用，键码为 14。

(5) F5：供用户使用，键码为 06。

(6) →：将光标右移一位，并把它经过的那些字符当做刚从键盘输入一样，送入键盘缓冲区。

(7) ←：将光标左移一位，并把它经过的那些字符从键盘缓冲区删掉。

(8) ↑：将光标上移一行。

(9) ↓：将光标下移一行。

(10) 中文：使计算机进入中文状态。

(11) 西文：使计算机进入西文状态。

7. 中华学习机几种工作状态是如何相互转换的

中华学习机的工作状态有很多种，有西文与中文工作状态，有浮点 BASIC (CEC-BASIC) 和整数 BASIC 工作状态，有监控工作状态和小汇编工作状态，有 LOGO 语言和其它高级语言工作状态工作状态。这里主要讨论浮点 BASIC、整数 BASIC、监控和小汇编四种工作状态之间的互相转换。

开机后中华学习机就自动进入浮点 BASIC 状态，它的标志为“]”。整数 BASIC 解释程序存贮在系统软磁盘中，文件名为 INTBASIC。要使中华学习机进入整数 BASIC 状态，需首先将解释 BASIC 程序存入机内 16K RAM 中，再键入：

] INT

即可。这时屏幕上出现提示符“>”。在 BASIC 状态下键入 CALL-151 命令可使计算机进入监控状态，监控状态的提示符为“*”。在监控状态下可以查看、修改、输入和运行 6502 机器语言程序。中华学习机 ROM 中固化有小汇编程序，它的作用是将 6502 汇编助记符翻译成 6502 机器指令。在监控状态下键入：

* D350G

即可进入小汇编状态。它的提示符为“!”。对于中华学习机无需调入 INTBASIC 解释程序就可直接进入小汇编状态。

这四种工作状态的相互转换可参看图 7-1。

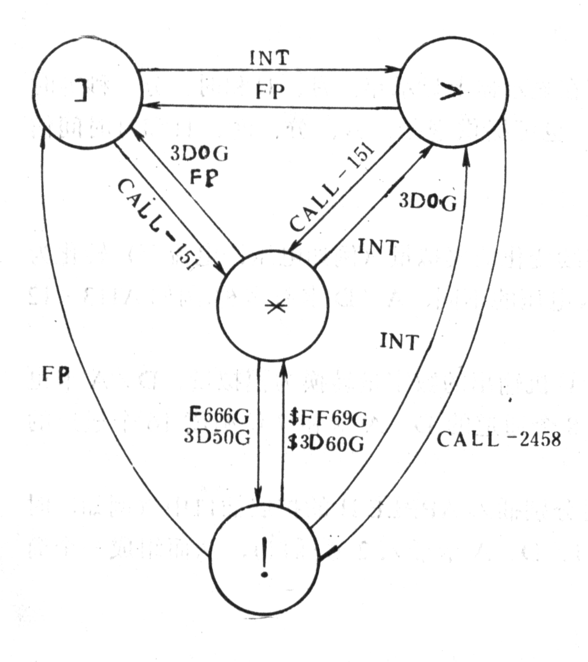


图 7-1 四种工作状态的相互转换

在使用各种命令时应注意以下几点：

1. 从监控状态转换到小汇编状态时应注意：如果是从浮点 BASIC 状态进入监控状态的，应使用 D350G 命令；如果是从整数 BASIC 状态进入监控状态的。应使用 F666G。

2. CTRL-B 表示同时按下 CTRL 键和 B 键，CTRL-C 表示同时按下 CTRL 和 C 键。它们能使计算机由监控状态回到原来转入监控状态前的那种 BASIC 状态。CTRL-B 会将内存中的 BASIC 程序和变量消除，而 CTRL-C 能保存内存中的 BASIC 程序和变量。按下 QUIT 键其功能与 CTRL-C 键一样。

3. 按 CTRL-RESET 键可使计算机由监控状态回到原 BASIC 状态，也可使计算机由小汇编状态回到 INTBASIC 状态。

4. 在小汇编状态下使用各种命令（包括监控命令）应注意在命令前加“$”。

8. 中华学习机有哪些常用的接口卡

中华学习机主机板上有一个槽口，可插入各种接口卡。插入接口卡后，可使主机与外部设备连接或扩展其功能。常用的接口卡有：

1. Z80 卡与打印机卡

插入 Z80 卡（Z80 soft card），可使中华学习机系统实现 6502——Z80 双 CPU 工作状态。可以引用 Z80 系统微机上使用的 CP／M 操作系统，扩展了中华学习机使用的高级语言和与 Z80 通用软件的兼容范围。

插入打印机卡，可以使用打印机打印程序、程序运行结果、图形、汉字等。

目前，为中华学习机设计的一种 Z80–打印机双功能卡具 Z80 卡与打印机卡的所有功能，而且价格很低。

2. 8088 卡

8088 卡上有 Intel8088 微处理器、2KRAM 和 4KROM。可使 APPLE Ⅱ微机系统实现 6502–8088 双 CPU 工作状态。可以使用 CP／M86 操作系统，使 APPLE Ⅱ与 IBM PC 微机在软件上实现兼容。

3. 时钟卡

时钟卡具有定时功能，插入此卡后，可在显示器上显示年、月、日和时、分、秒等时间信息。它可以计算 1 毫秒到 1 年的时间，也可提供毫秒、秒、分、时、日和月时间信息。

4. A／D 卡与 D／A 卡

A／D 卡是模—数转换器，它可以将连续变化的模拟量（例如电压、电流等）转化为数字量送至计算机内。根据转换精度与输入电压的不同，A／D 卡有多种，例如 AI13×12 位 A／D 卡等。用户可根据要求进行选用。

D／A 卡是数—模转换器，它可以将计算机输出的数字量转换为模拟量。D／A 卡也有许多种，常用的有 8 位精度 2 个通道到 8 个通道的 D／A 卡和 7 位精度 16 个通道的 A／D+D／A 卡等。

时钟卡、A／D 卡和 D／A 卡可以同时分别插入 APPLE II 的扩展插口中（例如，时钟卡插入 7 号插口、A／D 卡插入 4 号插口、D／A 卡插入 2 号插口），从而组成一个简单的计算机自动控制系统。

5. EPROM 写入卡

它具有对 EPROM 写入信息的功能、能自动写入、核对及空白测试，可用于复印或改写 EPROM 器件的内容。该卡又分为每次写一片和每次写八片两种。

6. 集成电路测试卡

该卡可以快速查出 TTL、CMOS 和 RAM 集成电路片的型号，并能循环检测集成电路片的稳定性。能检测 54／74 系列、54／74LS 系列、54／74S 系列、54／74L 系列及 54／74H 系列的集成电路片。

7. IEEE—488 卡

它可以使 APPLE Ⅱ微机能直接控制传统的测验仪表，和有关仪器构成一个新系统。

它可以控制多达 15 台的标准接口仪表。

8. RS—232C 串行接口卡

它可将计算机与智能外部设备接通（例如接通高精度图形输入板等），通讯速度为 75—19250 波特。它还可以作为多机联网的接口。

此外，APPLEⅡ微机还有 M6809 卡，可使机器执行 6809 软件；M68000 卡，使机器成为 16 位机，并可执行 68000 软件；调制电话卡，可处理电话信息；语言卡，可用编程控制扬声器发出语言信号，为人机对话创造条件；音乐卡，可通过编程控制扬声器发出高质量的音响效果。

9. 中华学习机常用的操作系统有哪些

目前，中华学习可以运行三种操作系统：DOS、CP／M 和 U、C、S、D PASCAL 操作系统。DOS 操作系统是中华学习机的基本操作系统，在此操作系统支持下，中华学习机可以运行 FPBASIC、INTBASIC 和 6502 机器语言程序。在 Z80 卡插入情况下，中华学习机可以运行 CP／M 操作系统，使中华学习机与 8080、8085 与 Z80 系统微型计算机实现部分软件兼容。在 CP／M 操作系统支持下，中华学习机可以运行 FORTRANIV、COBOL、ALGOL、MBASIC、GBASIC、Z80 汇编语言。中华学习机还可以在 U、C、S、D PASCAL 操作系统支持下，运行 PASCAL 和 FORTRAN 77 语言程序，以及与 16 位微机（例如 IBM−PC 机等）实现部分软件兼容。

10. 中华学习机有哪些常用的软件

1. 操作系统和语言软件

(1) DOS3.3 MASTER：该软件是 DOS3.3 操作系统主磁盘，具有 DOS 命令所需要的全部磁盘文件和示范实例。

(2) APPLE CP／M：该软件是 CP／M 操作系统主磁盘，具有与大部分操作命令有关的磁盘文件，同时还包括有 GBASIC 和 MBASIC 解释程序文件。

(3) CP／M FORTRAN−80：该软件为在 CP／M 操作系统管理下，能运行 FORTRAN−80 语言程序提供磁盘文件和子程序库。

(4) CP／M COBOL−80：该软件为在 CP／M 操作系统管理下，实现 COBOL 语言程序运行的磁盘文件和子程序库。

(5) CP／M ALGOL：该软件为在 CP／M 操作系统管理下，实现运行 ALGOL 语言程序提供的磁盘文件和子程序库。

(6) APPLE PASCAL 0～2（共三张软磁盘）：该软件包括有 PASCAL 操作系统命令所需要的磁盘文件和运行 PASCAL 语言程序所需的磁盘文件和子程序库。

(7) APPLE FORTRAN 1～2（共两张软磁盘）：该软件为在 PASCAL 操作系统管理下，实现运行 FORTRAN 77 语言程序所需要的磁盘文件和子程序库。

2. 系统诊断软件

(1) APPLE DOCTOR：该软件可用来检查中华学习机和 APPLE Ⅱ微机中主机的

RAM、ROM、键盘和 CRT 显示器的工作状态，以便分析故障的原因。

(2) APPLE CILLING Ⅱ：该软件可对主机中 RAM 和 ROM，RAM 扩展卡和磁盘驱动器进行检查测试。

(3) NIKROM MASTER DIAGNOSTIC PLUS：该软件具有较强的检测功能，可对中华学习机与 APPLEⅡ微机进行 RAM、ROM 驱动器和多种外部设备的工作状态进行检测。

(4) SYSTEM DIAGNOSTIC：该软件能检测主机 RAM、ROM、RAM 扩展卡、CRT 显示器和部分外围硬件卡。

(5) CP／M DIAGNOSTIC：该软件能检测 Z80 卡的功能和工作状态，检查 Z80 CPU。

3. 图形处理软件

(1) APPLE MECHANICS：该软件能完成几种图形处理工作。例如：高分辨率屏幕字符显示，修改磁盘中的图形数据结构以及产生多种高分辨率图形表等。该软件还存有多种图形花样子程序，可供用户作图时选用。

(2) SUPERPLOT：该软件是显示函数图象的软件，它允许用户键入多至五个数学函数，计算机可以把它们的图形一一显示出来。

(3) PRINT SHOP：该软件是打印图形的软件。它可以用计算机打印封面图案、贺年片以及用户绘制的图形。该软件建有许多图案组成的图库，并具有预检功能。

(4) APPLE WORD：该软件是一个能产生三维立体图形的软件包，可以建立三维立体图形，将图形移位、放大和缩小等。

(5) FANTAVISION：该软件是动画生成软件。只要用户给出初始图案和终止图案，图案就可以一个个连起来，做成动画系列。画图全部使用图案式菜单命令，使用方便，而且还可以连接游戏器。该软件还具有将图形放大、缩小、压缩、旋转及获得对称图形等功能。

(6) COMIC STRIP MAKER：该软件是连环漫画制作软件。它提供米老鼠、唐老鸭、果菲等多种角色，有 48 种姿态，15 种背景和 47 种道具，可供用户自已创作连环漫画。而且该软件还可以添加对话框和对话内容，存贮和打印用户设计的连环漫画。

(7) THE NEWSROOM：该软件是自行编辑打印的软件。它有 600 幅现成的图案，都能单独调用或配用，能任意置位、复制、涂色，并有大小五套英文字体供报头、标题和文本内容使用。组成的报纸可以存贮打印，并可通过调制解调器利用电话线作运程发送。

4. 数据管理与文字处理软件

(1) DBASE－Ⅱ：该软件称为关系数据库管理软件。它是在 CP／M 操作系统管理下的一个程序包，由 12 个磁盘文件组成。它有很强的数据管理和资料检索能力，广为使用。

(2) DB MASTER：该软件是处理文件的软件。它可用于建立、增删、插入、排序和检索文件，功能强，用途广。

(3) WORDSTAR：该软件是应用广泛的文字处理软件。利用它可以通过计算机处理文字资料，例如输入、修改、编辑、简单排版、复制和打印等。使用该软件需要有

80 列字符显示卡和 Z80 卡的硬件支持，并需要 CP／M2.2 系统主磁盘。

(4) APPLE CCHB：该软件为汉字处理系统软件。它在 CP／M 操作系统下，采用字型汉字编码输入，不需要专用汉字卡即可显示 6000 余个汉字。

(5) 中华超级汉字系统：该软件由二张磁盘组成（一个系统盘、一个字库盘），汉字系统功能较强，配有一、二级字库，可实现汉字打印。如配用 128K 扩展卡，可取代字库盘。

(6) STC 4.0 软汉字系统：该软盘由三张磁盘组成（系统主盘一张、编码一张和字库盘一张）。它配有一、二级字库，可采用电报、繁体仓颉、笔形、数形、国际等输入方式。它的功能齐全，可打印汉字，字体有宋体、仿宋体和繁写三种。

5. 系统功能扩展软件

(1) ASSEMBLER／EDITOR：该软件可在 DOS 操作系统下运行 6502 汇编语言程序。软件中的编辑程序（EDITOR）可将源程序用 ASC II 码送入计算机，再由汇编程序(ASSEMBLER) 将送入计算机的源程序翻译成目标程序并存入磁盘，然后计算机才能运行该 6502 汇编程序。

(2) ALDS (CP／M)：该软件是在 CP／M 操作系统下的汇编语言开发软件。可以使用 8080、Z80 和 6502 汇编语言。

(3) APPLE BOSS：该软件可在 DOS3.3 操作系统下，用于修改 DOS 命令和出错信息。

(4) DAKIGS AIDS：该软件含有 12 个应用程序，具有列程序清单、复制文件和磁盘产生变量或分支交互列表、修改磁盘内容、编辑数和打印显示内容等功能。

目前，中华学习机的软件大部分都是原 APPLE Ⅱ微机的软件，因为中华学习机与 APPLE Ⅱe 微机兼容，所以中华学习机可以运行这些 APPLE Ⅱe 的软件。

11. 中华学习机内存是如何分配的

中华学习机的微处理器是 6502，它有 16 根地址线，可寻址 64K 个内存单元(1K＝1024)。因此 CEC-I 型中华学习机中的 ROM、RAM、以及输入输出(I／O)设备所占用的存储空间，都分布在这 64K 地址的内存区域中。

在中华学习机中，所用的存贮空间大于 64K 内存单元，为此采用了存储体的切换技术，使同一逻辑地址空间可以映射到多个同样大小的物理地址空间。也就是说几块相同大小的存储体可以有同一的地址范围。通过设置存储体切换控制开关，控制逻辑地址到物理地址的映射，使得某一逻辑地址空间可以分时映射到多个同样大小的物理地址空间。也就是说可以在不同时间使用同一地址范围的不同的存储体。这样，大大地扩展了中华学习机的存储容量。

CEC-I 型中华学习机的内存分配如图 11-1 所示。图中 $ 0000～ $ BFFF 为主 RAM 空间，输入输出占用 $ C000～ $ CFFF 空间， $ D000～ $ FFFF 为存储体切换空间，此空间可以映射 RAM，也可以映射 ROM。主 RAM 空间占用 48K 地址，输入输出占用 4K 地址，存储存切换空间为 12K 地址。

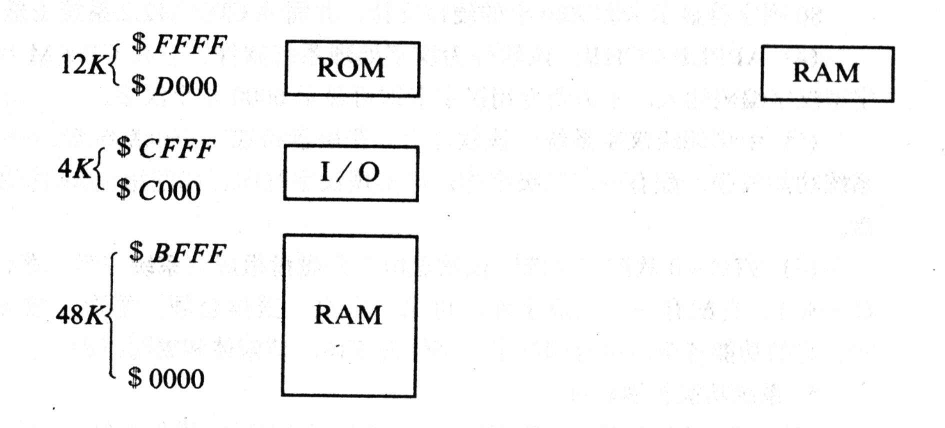


图 11-1

如果存储体切换空间是 12KROM 时，ROM 中可以存放 CEC-BASIC 解释程序、监控程序、小汇编程序、LOGO 解释程序，整数 BASIC 解释程序或 PASCAL 操作系统等。开机后，存储体切换的 12KROM 中存放的是 CEC-BASIC 解释程序、监控程序和小汇编程序。这时整个 64K 地址空间的分配如表 11-1 所示。

表 11-1　内存分配

地址	十进制	内存分配	
$ 0000	(0)	系统使用	
$ 0100	(256)	系统堆栈	
$ 0200	(512)	键盘输入缓冲区	
$ 0300	(768)	系统指针和供用户使用	
$ 0400	(1024)	文本和低分辨率图形第一页屏幕显示存贮区	
$ 0800	(2048)	文本和低分辨率图形第二页显示区	用户程序和各种变量及字符串数据
$ 0C00	(3072)		
$ 2000	(8192)	高分辨率图形第一页显示区	
$ 4000	(16384)	高分辨率图形第二页显示区	
$ 6000	(24576)		
$ 9600	(38400)	DOS	
$ C000	(49152)	I/O 区	
$ D000	(53248)	CEC-BASIC 语言解释程序和小汇编	
$ F800	(63488)	监控程序	
$ FFFF	(65535)		

12K 的存储体切换空间可以映射到 16KRAM 空间。这是因为地址空间 $ D000～ $ DFFF 可以映射到两个 4KRAM，即访问 16KRAM 时，要对 $ D000～ $ DFFF 空间访问两次，分时访问地址空间的两个 4KRAM 存储空间。如图 11–2 所示。

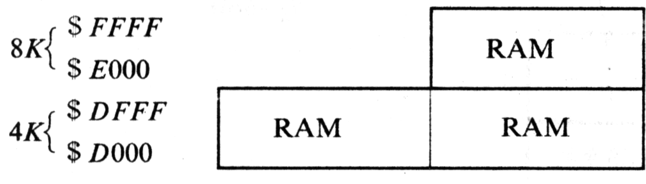


图 11–2

存储体的切换是通过改变软开关的状态来实现的。在此空间进行存储体切换，要通过设置软开关的状态来做三个选择：一是该空间是映射到 RAM 还是 ROM，二是允许写 RAM(同时也可以读)还是禁止写(只读)RAM，三是地址 $ D000～ $ DFFF 是映射到第一个 4KRAM 还是映射到第二个 4KRAM。存储体切换软开关地址如表 11–2 所示。

表 11–2　存储体空间切换软开关地址

软开关地址			读	读	4KRAM		注
十六进制	十进制	RAM	RAM	ROM	第一体	第二体	
$ C080	49280		·			·	
$ C081	49281	·		·		·	1
$ C082	49282			·		·	
$ C083	49283	·	·			·	2
$ C084	49284		·			·	
$ C085	49285	·		·		·	1
$ C086	49286			·		·	
$ C087	49287	·	·			·	2
$ C088	49288		·		·		
$ C089	49289	·		·	·		1
$ C08A	49290			·	·		
$ C08B	49291	·	·	·			2
$ C08C	49292		·		·		
$ C08D	49293	·		·	·		1
$ C08E	49294			·	·		
$ C08F	49295	·	·		·		2

在使用上述软开关时应仔细筹划，否则会破坏程序。在使用时应注意：

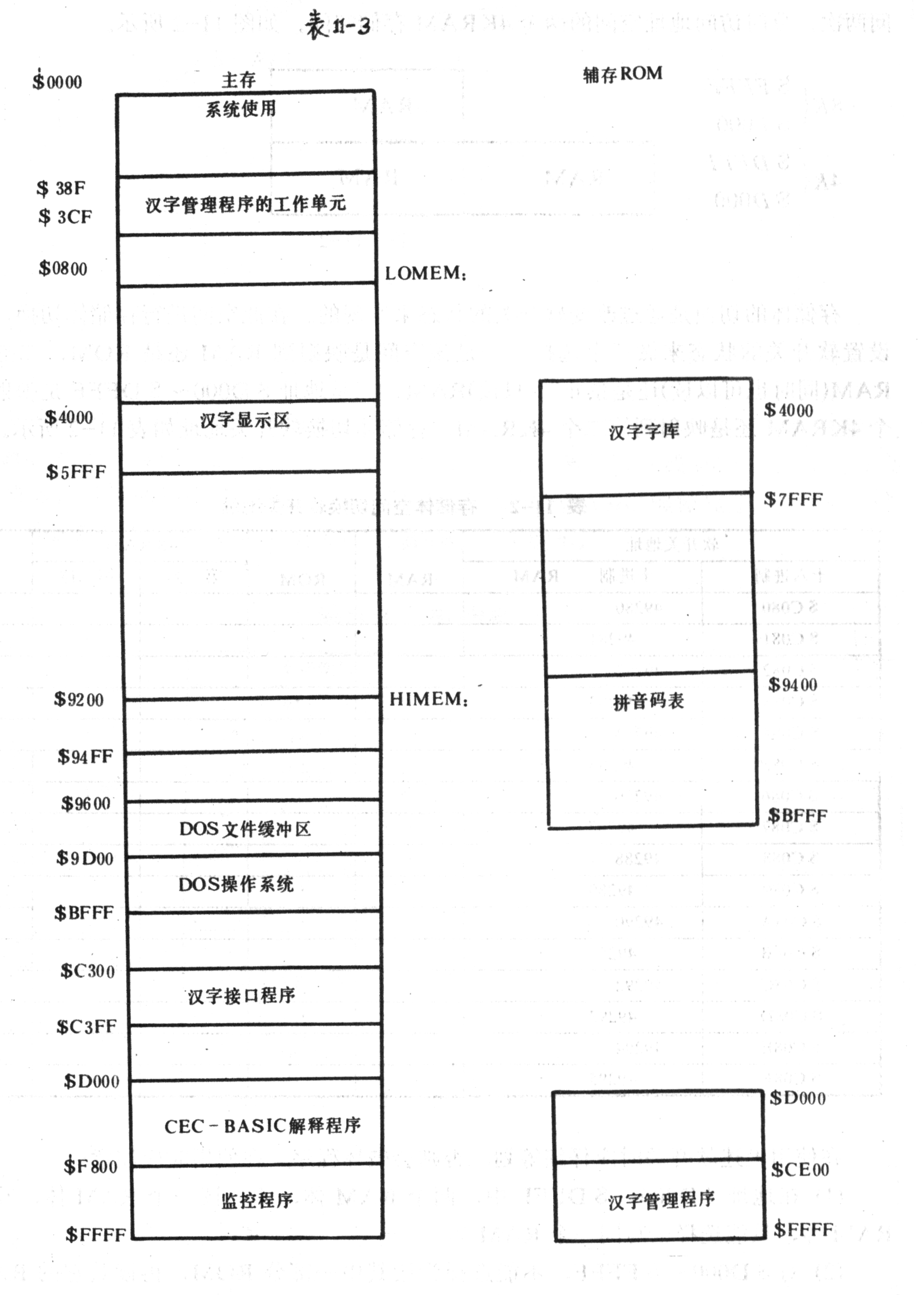
(1) 在地址 $ D000～ $ DFFF 中，两个 RAM 体，不能读一个 RAM 体，写另一个 RAM 体，只能选择读写同一个 RAM 体。

(2) 对 $ D000～ $ FFFF，不能选择为读其中一部分 ROM，再读其它的 RAM。例如，不能读 ROM 中的监控程序的同时再读该空间的 RAM。若要使该空间 RAM 中的程序能调用监控程序，必须将监控程序拷贝到 RAM 中。

CEC–I 型中华学习机的汉字系统占用了辅存 ROM 空间和部分主存 RAM 空间。辅

存 ROM 占用的地址空间为＄4000～＄BFFF 及＄D000～＄FFFF。同样也采用存储体切换技术来选用辅存 ROM。汉字系统下内存分配情况如表 11-3 所示。

表 11-3



12. 如何使用中华学习机的汉字系统

中华学习机的内存 ROM 中固化有汉字管理系统和 6763 个常用的一、二级汉字字库。使用中华学习机的汉字系统包括汉字系统的进入与退出，汉字的输入，编写中文 BASIC 语言，写汉字文件，打印汉字等。这里主要介绍如何进入和退出汉字系统以及如何输入汉字。其它的使用方法将在以后的问题中答复。

1. 如何进入汉字系统

(1) 直接按下“中文”键即可进入汉字状态。它适用于 BASIC、监控或 DOS 操作系统三种状态。

(2) 在监控方式下，可使用命令 3 CTRL−P，就是先键入 3，再按 CTRL−P，最后按回车键。

(3) 在 BASIC 或 DOS 操作系统状态下，可使用命令 PR#3 进入汉字状态。在 BASIC 程序中使用 PR#3，必须在该命令后面跟一条 PRINT 语句。

进入汉字状态后，屏幕显示处于高分辨率图形第二页，HIMEM: 指针指向内存地址 $ 9200。内存中的 BASIC 程序如不太长的话，不会受到破坏。

2. 如何退出汉字系统

(1) 直接按下“西文”键即可由中文状态退出。它适用于 BASIC、监控或 DOS 操作系统三种状态。

(2) 在监控方式下，可使用 C33AG 命令。

(3) 在 BASIC 程序中，可使用 TEXT 或 PRINT CHR $ (17) 命令。

(4) 在 BASIC、监控或 DOS 操作系统状态下，按 CTRL−RESET 或 CTRL−Q 都可以退出中文状态。

退出中文状态后，屏幕显示回到文本第一页，HIMEN: 指针指向仍为 $ 9200。

3. 汉字输入方式的选择

CEC−I 型中华学习机汉字输入方式有拼音输入方式与区位码输入方式。按下“F2”键或按下 CTRL−L 后，在屏幕最下边一行(状态行)左边显示“拼音”两字，这时就可以采用拼音输入方式输入汉字了。按下“F3”键或按下 CTRL−W 后，在屏幕状态行左边显示“区位”两字，这时就可以采用区位输入方式输入汉字。按下“F1”键或 CTRL−A，即可使计算机由汉字输入状态回到字母输入状态，在屏幕状态行左边将显示出“字母”两字。

按下 CTRL−Q 键可使控制状态提示符由显示变为不显示，如果再按一下 CTRL−Q 键又可使控制状态提示符重新显示出来。

4. 拼音输入方法

拼音输入方式是使用 26 个字母作为输入码，键入的码长为 1～6 码，大小写字母均可输入。输入时按声母、韵母依次键入。键入第一个字母时，屏幕状态行会显示出该声母开头的音节中最常用的 6 个汉字供用户选择，如果有要输入的汉字，则只要打入相应的数字，被选的汉字就出现在光标所存处；如果没有要输入的汉字，就要继续打入声母或韵母。例如“深”字的声母为“sh”，韵母为“en”。先键入“s”，则状态行显示为:

拼音：s　　　　　　　　1 所　2 三　3 思　4 四　5 色　6 速

状态行右边没有"深"字，再按第二个字母"h"，状态行显示：

拼音：sh　　　　　　　　1 是　2 上　3 时　4 生　5 说　6 社

仍没有"深"字，继续按第三个字母"E"，状态行显示：

拼音：she　　　　　　　　1 砷　2 赊　3 蛇　4 舌　5 舍　6 赦

仍没有"深"字，按第四个字母"N"，状态行显示：

拼音：shen　　　　　　　　1 砷　2 申　3 呻　4 伸　5 身　6 深

状态行有"深"字，按相应的数字键 6，"深"字即在光标处显示出来。

当键入所有声、韵母后，状态行仍没有所需的汉字，可按"＞"键，每按一次，状态行会显示出下一幕的 6 个同音汉字，直到计算机发出"嘟"声为止，表示该音节的同音汉字全部提示完。例如"沈"字，在按完 shen 字母后提示行仍没有"沈"字，这时按一下"＞"键状态行显示：

拼音：shen　　　　　　　　1 娠　2 绅　3 神　4 沈　5 审　6 婶

按一下数字键 4，沈字即出现在光标所在处。

在输入字母时，如果有的拼音字母打错了，可以直接用"◁"键删除状态行上最后一个拼音字母，若状态行上的拼音字母都删除了，再按"◁"键，则将删除光标所在处的汉字或其它字符。如果按空格键，可一次将所有拼音字母全部删除。

5. 区位码输入法

区位码输入方式是使用国家标准 GB2381−80 区位方式输入汉字。在输入汉字时，仅使用 0～9 共 10 个数字作为汉字输入码，通过敲入四位数字可以将国家标准中规定的汉字或图形字符输入到计算机中。例如，"吐"的区位码是 4534，键入 4534 后，"吐"字就在光标所在处显示出来。

国家标准规定，高位两个数字是表示区位码，低两位数字表示位码，00−10 区为图形字符 16−55 区为国标一级汉字，56−87 为国标二级汉字，每个区有 94 个汉字或字符，其位号为 01−94。

在使用区位码输入汉字时，除 0～9 数字键外，键盘上其它字符均可直接输入计算机，在光标所在处显示出来。

在汉字输入过程中，如果有一个数字敲错了，可用"◁"键来删除。如果键入的四位数字区位码不在国标码所规定的范围内，机器将发出一声"嘟"以示警告，用户应重新输入或删除。

由于区位码不易记忆，所以只有在拼音方式下查找不到的汉字，才使用区位码方式进行输入。另外，区位码方式还常常用来制表格或输入一些其它特殊符号。

6. 特殊符号的输入

使用区位码输入方式，可输入英文字母、汉语拼音字母、日文假名、俄文字母、希腊字母、拉丁字母和其它一些特殊的字符。在拼音输入方式下，也可输入一些特殊符号。键入一(减号)、\(斜线)或＝(等号)后，在状态行会显示一些字符供用户选用(需通过按“>”、“<”键来挑选)。选中所要字符后，只要按字符的编号，字符就会显示在光标所在处。

13. 中华学习机是如何使用零页内存单元的

中华学习机的微处理器(CPU)采用的是6502，在6502的寻址方式中，有几种方式需要使用零页内存单元空间$0000～$00FF。使用零页寻址方式具有使用灵活和占时间短的特点，所以BASIC解释程序、监控程序和DOS都大量使用零页，以提高程序的效率。

浮点BASIC解释程序占用的零页内存单元是：$0000～$0005、$000A～$0019、$001D、$0050～$00CD、$00D0～$00D5、$00D8～$00EA、$00F0～$00F8。整数BASIC解释程序占用的零页内存单元是：$004A～$004D、$0055～$00DF、$00FE～$00FF。监控程序占用的零页内存单元是：$0000～$0001、$001F～$0049、$004E～$0055。DOS 3.3占用的零页内存单元是：$0026、$0027、$002A～$002F、$0035～$0039、$003E～$0048、$004A～$004D、$0067～$006A、$006F、$0070、$00B0、$00CA～$00CD、$00D8。

零页内存单元使用情况简述如下：

内存单元地址	说明
$00～$05	继续CEC-BASIC的转移指令。
$0A～$0C	USR函数的跳转指令地址。
$0D～$17	通常用于CEC-BASIC的计数器／标志。
$20～$4F	监控程序的保留单元。
$50～$61	CEC-BASIC的指针。
$62～$66	上次乘法／除法的结果。
$67～$68	指向程序开始的指针。
$69～$6A	指向简单变量区域开始位置的指针，即程序的结尾地址，并加上1或2。LOMEM：语句可修改它的内容。
$6D～$6E	指向使用中的数值存储区域的末端指针。
$6F～$70	指向字符串存储区域开始的指针，字符串从此开始一直存储到内存末端。
$71～$72	通用指针。
$73～$74	CEC-BASIC可用的内存的最高单元加1。在最初进入CEC-BASIC

时，置成最高的可用的 RAM 内存地址。

$ 75～$ 76	当前正在执行的程序行行号。
$ 77～$ 78	“旧行号”通过CTRL−C、STOP或END语句来建立的，它给出程序执行时被中断的行的行号。
$ 79～$ 7A	“旧的正文指针”指出下一次将要执行的语句在存储器中的地址。
$ 7B～$ 7C	当前DATD语句的行号，该行中的数据正由READ语句来读。
$ 7D～$ 7E	正在读的数据所在内存单元地址。
$ 7F～$ 80	指向当前输入的源程序的指针，在INPUT语句执行期间置为 $ 201(第二页为键盘缓冲区)，在执行一条 READ 语句时，被置为程序正读的 DATA。
$ 81～$ 82	保留上次使用的变量名。
$ 83～$ 84	指向上次使用变量的值的地址。
$ 85～$ 9C	通用。
$ 9D～$ A3	主要的浮点累加器。
$ A4	通常用于浮点的数字例行程序。
$ A5～$ AB	次要的浮点累加器。
$ AC～$ AE	通常用于标志／指针。
$ AF～$ B0	指向程序的末尾（不能用LOMEM：改变）。
$ B1～$ C8	CHRGET例行程序。CEC−BASIC每次希望得到另一个字符时，在这儿调用它。
$ B8～$ B9	指向通过CHRGET例行程序得到的上一个字符的指针。
$ C9～$ CD	随机数。
$ D0～$ D5	高分辨率绘图的画线指针。
$ D8～$ DF	ONERR指针／画线指针。
$ E0～$ E2	高分辨率绘图的X和Y坐标。
$ E4	高分辨率绘图的颜色代码。
$ E5～$ E7	通用于高分辨率绘图。
$ E8～$ E9	指向图形表开头的指针。
$ EA	高分辨率绘图的碰撞计数器。
$ F0～$ F3	通用标志。
$ F4～$ F5	ONERR指针。

二、 中华学习机 BASIC 语言与汇编语言

14. 中华学习机可以使用的 BASIC 语言有哪几种

中华学习机在 DOS 操作系统支持下，可以使用浮点 BASIC 和整数 BASIC 语言。浮点 BASIC 也叫 CEC–BASIC 或实数 BASIC，可用 FPBASIC 或 APPLE SOFT BASIC 表示，它的解释程序固化在主机板的 ROM 中。整数 BASIC 可用 INTBASIC 表示，它的解释程序在 DOS 操作系统磁盘中，使用前应将它调入主机 RAM 中。浮点 BASIC 语言处理的范围较大，适用于科学计算、统计管理；整数 BASIC 语言处理一定范围内的整数，运算速度快，占内存少，适用于画图、游戏和辅助教学。两种 BASIC 都是西文 BASIC。中华学习机还能使用中文 BASIC。中文 BASIC 所用的语言、命令，不论书写格式还是其功能都与浮点 BASIC 和整数 BASIC 基本一样，只是个别语句中文 BASIC 不能使用。中文 BASIC 只是在进行字符串联接、注释和建立文本时使用中文。

在 CP／M 操作系统支持下，中华学习机还可以使用 MBASIC 和 GBASIC 两种语言。在 CP／M 系统主磁盘中存有这两种语言的解释程序，它们都是 MICROSOFT 软件公司为 APPLE Ⅱ微机编制的 MBASIC 语言属于一种扩展 BASIC 语言，而 GBASIC 语言是在 MBASIC 语言基础上附加了绘图语句，具有更强的功能。

15. 如何输入和执行各种 BASIC 语言程序

1. 如何输入和执行 FP BASIC 语言程序

当中华学习机处于浮点 BASIC 状态(屏幕上出现提示符“]”)时，即可直接输入 FPBASIC 语言程序，并可用 RUN 命令直接执行。例如:

```
10  A=5: B=4: C=3
20  PRINT A; "+"; B; "+"; C; "="; A+B+C
RUN
5+4+3=12
```

2. 如何输入和执行 INTBASIC 语言程序

当中华学习处于整数 BASIC 状态(屏幕上出现提示符“＞”)时，即可直接输入 INTBASIC 语言程序，并可用 RUN 命令直接执行。例如:

```
>  AUTO 10, 5
>  10 FOR I = 0 TO 3
>  15 S = S+I
>  20 NEXT I
```

```
>  25 PRINT"S=";S
>  30 END
>  35 CTRL-X
>  MAN

>  RUN
   S=6
OK
```

这里 AUTO 命令是用置自动行号方式的。使用此命令后，机器能自动给出行号。AUTO 后面第一个数给出起始行号，第二个数给出行号间隔。在执行程序前，必须退出自动编号方式。退出方法是键入 CTRL-X 再键入 MAN 命令。

3. 如何输入和执行 MBASIC 语言程序

在插入 Z80 卡扩展卡情况下，将 CP／M 系统主磁盘放入磁盘驱动器中，开机后中华学习机进入 CP／M 操作系统，屏幕上出现如下提示符：

```
APPLE Ⅱ CP/M
56K VER 2.20B
(C) 1980 MICROSOFT
A>
```

然后用 DIR 命令将主磁盘文件目录列表如下：

```
A>DIR
A: FORMAT      COM: COPY        COM
A: MBASIC      COM: GBASIC      COM
A: CONFIGIO    BAS: PIP         COM
A: STAT        COM: ED          COM
A: ASM         COM: DDT         COM
A: LOAD        COM: RWB         COM
A: APDOS       COM: SUBMIT      COM
A>
```

接着键入

```
A>MBASIC
```

屏幕上出现下列提示符：

```
BASIC-80 REV.5, 2
[ APPLE CP/M VERSION ]
COPY RIGHT (C) 1980 BY MI CROSOFT
CREATED: 12-NOV-80
26483 BYTES FREE
```

```
OK
```

于是可输入 MBASIC 语言程序，并用 RUN 命令执行程序。举例如下:

```
OK
AUTO  10,10     (进入自动编行号方式)
10  FOR I=1 TO 2 STEP 1
20  K=K+I
30  LPRINT K
40  NEXT I
50  ^C (CTAL-C, 即退出自动编行号方式)
OK
RUN
1
3
```

4. 如何输入和执行 GBASIC 语言程序

在列完 CP／M 主磁盘文件目录后，键入：A>GBASIC，屏幕上出现下列提示符:

```
BASIC-80 REV 5, 2
[ APPLE CP/M VERSION ]
COPYRIGHT (C) 1980 BY MICROSOFT
CREATED: 12-NOV-80
17393 BYTES FREE
OK
```

于是在 OK 提示符下可输入 GBASIC 语言程序，并用 RUN 命令执行程序。举例如下:

```
10  FOR A=0 TO 4 STEP 2
20  B=B+A
30  NEXT A
40  LPRINT "B="; B
OK
RUN
B=10
OK
```

16. 什么叫立即执行方式、什么叫延迟执行方式

1. 立即执行方式

如果从键盘上输入不带行号的语句或命令(对于 FPBASIC 和 INTBASIC 语言，可以键入多条语句，语句间必须用“:”隔开)，则在按 RETURN 键后，语句或命令会立即执行。这种执行方式叫立即执行方式。采用这种方式执行过的语句和命令不会再在内存中找到，LIST 命令不会将它们列出来。如果立即执行的语句使某些变量取得数据，则这些变量和它们所代表的数据，将保留在内存中。举例如下:

```
] LET A=4+3: PRINT A
7
PRINT A
7
```

2. 延迟执行方式

若将若干个程序行输入计算机，而每个程序行之首行号，则在每输入一个程序并以 RETURN 键为结束时，程序被送入内存中，并没有被执行。只有在键入 RUN 命令后，程序才会被执行。这种执行方式叫延迟执行方式。用这种方式执行后，源程序仍留在内存中，可以用 LIST 命令将其列出来。使用 RUN 命令执行一个程序时，首先会将内存中原有的变量所表示的数据清除。

17. 有哪些语句限于立即执行方式、有哪些语句限于延迟执行方式

整数 BASIC 语句

1. 限定用于立即执行的语句

AUTO　CLR　CON　DEL
HIMEN:　LOAD　LOMEN:　MAN
NEW　RUN　SAVE

2. 限定用于延迟执行的语句

END　FOR　GOSUB　INPUT
NEXT　RETURN

浮点 BASIC 语句

限定用于延迟执行的语句有:

DATA　DEF FN　GET　INPUT
ON ERR GOTO　RESUME

18. FPBASIC 和 INTBASIC 语言有何不同点

FPBASIC 和 INTBASIC 语言有以下不同点，除此之外，两种语言基本相同。

一、 常量、变量、函数与运算符。

1. 常量：数值型常量只能取−32767～32767 之间的整数，不能用科学计数法表示。使用字符串常量时，必须先定义字符串长度。

2. 数组：数组只能是一维的。而且必须先说明，后使用。不存在字符串数组。

3. 变量：组成变量的字符总数不得超过 100 个。它不是只识别前两个字符，而是识别所有组成变量的字符，因此可以大大扩充变量的个数。

4. 函数：它的算术函数只有 ABC(X)、SGN(X)、RND(X)三个。RND(X)给出 0～X(不含 X)之间的一个随机整数。字符串函数只有 ASC(X)和 LEN(X)两个。

除此以外还有 SCRN(X,Y)、PDL(X)、PEEK(X)、TAB(X)函数。PEEK(X)函数中的 X 取值范围为−32767～32767。TAB(X)函数中 X 的取值范围为 1～40。

5. 运算符：运算符多了 MOD 和#运算符。MOD 是模运算，它给出两数相除的余数。例如：

13 MOD 7＝6,　　5 MOD 4＝1.

#是不等运算符。对字符串作不等比较时需用#运算符，不能用＜＞运算符。

表达式的运算次序为：

(　　)

NOT ＋ － (正、负号)

× $ ＊ ／ MOD

＋ － (加、减号)

＝ (赋值号)

＝、 #或＜＞、＞、＜、＞＝、＜＝

AND

OR

二、 语句、命令和程序

1. 一个程序行不得超过 150 个字符。

2. 没有 READ／DATA 语句和 RESTORE 语句。

3. 没有高清晰度图形显示方式，因此也没有高分辨率绘图语句。

4. 程序必须以 END 结束。

5. 不能使用下列保留字，也就是不能使用这些保留字表示的语句和命令：

ATN、CHR$、COS、DATA、DEF FN、DRAW、EXP、FLASH、FN、FRE、GET、HCOLOR＝、HGR、HGR2、HOME、HPLOT、INT、INVERSE、LEFT$、LOG、MID$、NORMAL、ON...GOSUB、ON… GOTO、ONERR…、GOTO、POS、READ、RECALL、RESTORE、RESUME、RIGHT$、ROT＝、SHLOAD、SIN、SPC、SPEED＝、SQR、STOP、STR$、TAN、USR、VAL、WAIT、XDRAW、&

6. INTBASIC 语言使用的特殊语句和命令

(1) AUTO 命令，其命令格式为:

AUTO 数 1，数 2

这个命令可用于自动置行号。使用此命令后，用户输入程序时不用敲入一个个行号，而由机器自动以等间隔的方式提供使用。命令中数 1 为程序的起始行号，数 2 为程序间隔。当可选项省缺时，机器自动以数 2 为 10 处理。

(2) MAN 命令，其命令格式为:

MAN

它是一个撤消 AUTO 的命令，它前面必须加 CTRL−X 命令。只有在退出自动编行号方式后，才可以运行程序或执行立即命令。

(3) DSP 语句，其格式是:

[行号] DSP 变量名

这是一个置检查方式的语句。它的功能是：当给定的变量的取值发生变化时，连同行号一起显示出来。对数组元素，只能给出数组名，机器自动按下标当时的值取相应元素显示。

RUN 命令和 NO DSP 命令可以清除 DSP 方式。

(4) CLR 命令，其命令格式为:

CLR

它相当于 FBASIC 语言中的 CLEAR 命令。它可以使数值型变量全部清为 0，数组维数和字符串变量的长度清为 0。

(5) CON 命令，其命令格式是:

CON

它相当于 FPBASIC 语言中的 CONT 命令，可用来继续运行被 CTRL−C 中止的程序。

7. INTBASIC 与 FPBASIC 语言功能相近的语句和命令

INTBASIC 语言中的 INPUT、PRINT、CALL、POKE、DIM、DEL、TRACE、FOR...TO [STEP...] ...NEXT、IF...THEN、GOTO、GOSUB...、RETURN 等语句和命令与 FPBASIC 语言中相应的语句和命令稍有不同。

(1) INPUT 语句：机器要求输入字符串时，屏幕上仅有闪动的光标，而且要求每输入一个字符串后都要按一下 RETURN 键。机器要求输入数值时，屏幕只显示单问号提示符。INPUT 语句可以带提示字符串，但提示字符串之后应紧跟逗号而不是分号。

(2) PRINT 语句：屏幕一行分 5 个区域，每一个区域显示 8 个字符。它不能用“?”代替。

(3) CALL、POKE 语句：其命令中的地址取值范围为−32767～32767。

(4) DIM 语句：使用数值以前必须用 DIM 语句加以说明，使用字符串以前必须用 DIM 语句说明其长度。

(5) DEL 命令：只能用于立即执行方式。

(6) TRACE 命令：当一个程序行有多条语句时，它只显示一个行号。

(7) FOR...TO、STEP、NEXT 语句：要求初值、终值为整数，步长值为正整数，

NEXT 后面变量名不得省去。循环语句只能用于延迟执行方式。

(8) IF...THEN 语句：它的执行方式是，当条件不满足时，不是直接转至下一个程序行，而是跳过 THEN 后面的第一条语句，执行后续语句，然后再转至下一个程序行去执行。例如：

```
10  INPUT  N
20  IF  N = 1  THEN  PRINT  "* * * * * * * *":  PRINT "%%%%%%"
30  PRINT  "$ $ $ $ $ $"
40  END

RUN
? 1
* * * * * * * *
%%%%%%
$ $ $ $ $ $

RUN
? 2
%%%%%%
$ $ $ $ $ $
```

(9) GOTO、GOSUB 语句：这两条语句后面的行号可以由算术表达式给出。

三、 错误信息标志 1. INTBASIC 语言错误信息的表示格式为：

立即执行方式： ? X X ERROR

延迟执行方式： ? X X ERROR IN YY

其中"YY"为出错语句的行号。

2. FPBASIC 语言错误信息的表示格式为：

立即执行方式： * * * XX ERR

延迟执行方式： * * * XX ERR

19. 什么叫程序的优化

用计算机语言解决一个问题，可以编写许多不同的程序，这些程序不可能完全一样，自然有优劣之分。程序的优化就是使编写的程序尽量好。所谓一个好程序，应体现以下几个方面：

1. 程序结构合理、清楚，容易阅读和使用。
2. 程序执行效率高，运行时间短。
3. 程序占用的内存少。

以上各条相互联系、彼此牵制，这就要求编程者根据问题的性质，精选解决问题的方

法，综合其利弊，按不同要求，择优选取。例如，要让计算机做许多工作时，要特别注意程序运行时间要短；当计算机内存少，而程序较长时，应特别注意节省内存。

20. 如何使程序结构优化

所谓程序结构，是指程序中指令的组织方法。对于 BASIC 语言程序，是指如何编排 BASIC 语句。

一个程序结构不管如何复杂，它最终都是由一些最基本结构所组成。程序的基本结构分为顺序、选取和重复三种结构，如图 20-1 所示。这三种结构遵循一个共同的特征：结构只有一个入口和一个出口。图中各模块也可以是这三种结构中的一种。

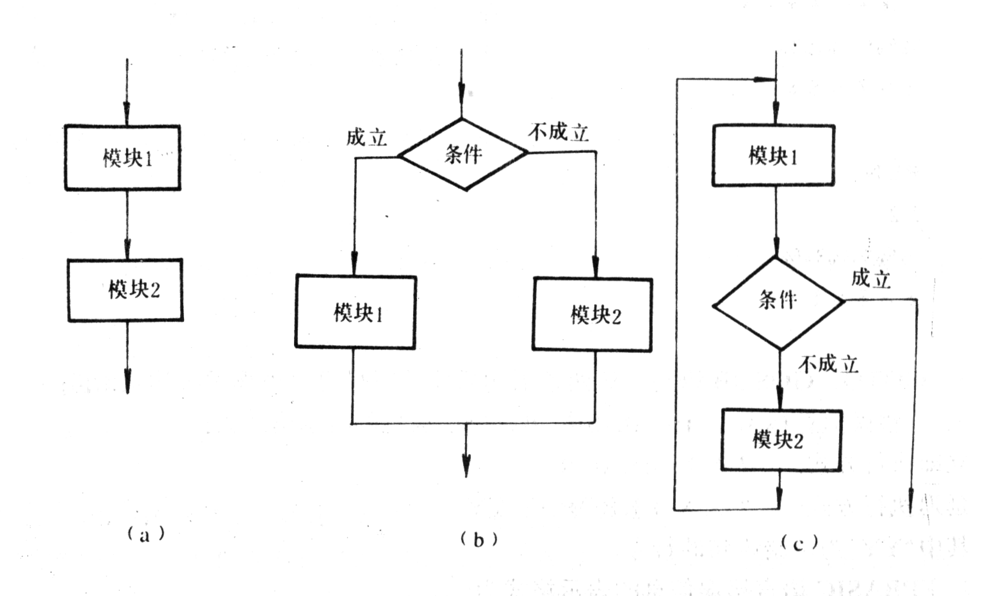


图 20-1

顺序结构表现为一串语句序列，程序运行时，顺序执行一连串语句。

选取结构表现为各种转向语句，程序运行时，根据条件是否成立来决定执行哪个模块。这种结构在 BASIC 语言中可用条件转向语句来实现。

重复结构是闭型结构，结构的出口被控制返回到同一结构的入口处，如此循环若干次后才能从结构中退出。显然，在 BASIC 语言中，可以用 FOR / NEXT 循环语句来构成这种结构，也可以用无条件和条件转向语句来构成。

程序结构优化是指程序结构合理和清晰，即结构模块化和程序的逻辑结构与问题的逻辑结构相对应。具有模块化的程序，便于阅读、修改和扩充。

模块化结构就是将一个长的程序分成适当规模的若干程序段，各程序段相对独立，每个程序段构成一个模块，有自己的入口和出口。再依据问题的特点将各模块按一定的逻辑关系结合到一起。

21. 如何使 BASIC 程序占用的内存空间少

为使 BASIC 程序占用的内存空间少，需要注意以下几点:

1. 一个程序行写多条语句: 每个程序行号要占用 5 个字节的内存，因此应尽量节省使用程序行，为此，可以在一个程序行中写多条语句，各语句间用“:”号隔开。但应注意一个程序行的字符个数不得大于 239 个。采用这种方法的缺点是给阅读和修改带来困难。

2. 删除注释语句: 这种方法会给阅读程序带来困难。一般应使注释语句尽量简洁。

3. 尽量用整型量: 一个整型量只占用两个内存字节，而实型量需占用五个内存字节，故尽量使用整型量可以节省内存。

4. 尽量用变量而少用常量。

5. 不写 END，在 FP BASIC 语言中，不需用 END 来结束程序，故可不用 END 语句。

6. 尽量使用程序前面已经用过的变量名。

7. 使用数组中下标为 0 的下标变量。

8. 少采用 LET 和 READ / DATA 语句赋值，可尽量选用 INPUT 语句或从数据文件中读取数据的方法赋值。

9. 打印语句中 TAB 函数前后的分号可省去，在能够分辨出各项意义的前提下，PRINT 语句中的分号也可以省去。

10. 把程序中多次出现的程序段编成子程序，用自定义函数定义多次使用的表达式。

11. 使用机器内部函数，这比用 BASIC 语言编写具有相同功能的程序占用内存少。

12. 省去 NEXT 后面的变量名。

13. 选择简洁的解决问题的方法。

22. 如何减少程序的运行时间

为了减少程序的运行时间，应注意以下几点:

1. 尽可能使用整型量不用实型量。

2. 能在循环语句外面执行的语句，尽量不要移到循环体内来执行。

3. 在程序中尽可能使用开关语句。

4. NEXT 后面的变量尽可能省去。

5. 尽量使用在程序前面用过变量。

6. 选择好的计算方式，使用简单的算术运算。计算机进行加法、减法运算比执行乘法运算快，执行乘法运算又比除、乘幂运算快。

7. 避免重复求值，重复进行的运算，可一次运算后将结果赋给一个变量保存。

8. 不使用注释语句。

9. 在程序执行时，若遇到一个新的行号，例如 GOTO500，则机器会从最低的行号到此行号为止扫描寻找一遍，再转至此行（500 语句行）去执行程序。因此，常用的子程

序和常用的程序行，应编写在程序的开始处，以减少扫描寻找的时间。

10. 在 BASIC 程序执行中先使用和常使用的变量应安排在变量表前面，以便节省每次寻找的时间。例如，语句：

10 LET A=3: LET B=4: LET C=2

执行后，变量 A 放在变量表首位，其次是变量 B，再次找到 A，再找到 B，最后找到 C。

23. 如何调试程序

一个新编好的程序，往往不能完全象用户所期望的那样马上就能工作，得到预期的结果。即使程序不存在 BASIC 语法方面的错误（这一点机器会自动指出）但在程序逻辑上却很可能有错误。将程序在机器中运行，同时寻找和清除程序的错误叫调试。调试常采用如下的方法：

1. 采用 TRACE 命令

TRACE 命令的作用是：在程序执行过程中，它可以使屏幕显示出先后执行的程序行行号，从而跟踪程序运行的流程。NOTRACE 命令可以解除上述的跟踪方式。下面通过一个例子说明 TRACE 命令的工作情况。

```
5     N =10
10    FOR I= 1 TO 2
20    FOR J= 1 TO 2
30    N = N+1
40    A (I, J) =N
50    NEXT J
60    NEXT I
70    FOR I= 2 TO 1 STEP -1
80    FOR J= 2 TO 1 STEP -1
90    PRINT A  (J, I),
100   NEXT J
110   PRINT
120   NEXT I
] RUN
14             12
13             11
] TRACE
] RUN
#5 #10 #20 #30 #40 #50 #30 #40 #50 #60 #20 #30 #40#50#30 #40#50
#60 #70 #80 #90 14 #100 #90 12
#100 #110
#120 #80 #90 13 #100 #90 11
```

```
#100 #110
#120
] NOTRACE
] RUN
14              12
13              11
```

2. 采用 GET 语句 GET 是一条 Applesoft 语句，执行此语句时，程序停止运行等待用户通过键盘送入一个字符，并将字符赋给 GET 后面的变量，然后程序往下顺序执行。在程序因执行 GET 语句而停止运行和用户键入字符时，屏幕不显示“?”和键入的字符。在程序调试过程中，可以利用 GET 语句给程序运行过程加暂停，以便观察程序各段的运行结果和程序的运行过程。举例如下：（程序运行结果见右）

```
10  FOR I =1 TO 10
20  PRINT TAB  (15 - I);
30  FOR J =1 TO I * 2 - 1
40  PRINT "*";
50  NEXT J
60  PRINT
70  NEXT I
] RUN
```

```
         *
        ***
       *****
      *******
     *********
    ***********
   *************
  ***************
 *****************
*******************
```

如果加入：45 GET A $，则可以看出每次到暂停处（45 语句）打印一个“*”。如果加入：55 GET A $，则可以看出每次到暂停处（55 语句）打印出一行“*”。

3. 采用 STOP 语句

在程序中加入 STOP 语句，会使程序暂停在 STOP 语句行。这时用户可以用立即执行方式检查程序中各变量值的变化情况。如要使程序由暂停处往下顺序执行以后的程序，可键入 CONT 命令。举例如下：

```
5   A (0, 0) = 1: A (0, 1)  = 1
10  FOR I = 1 TO 2
20  FOR J = 1 TO 2
30  A  (I, J) =A (I - 1, J) +A  (I - 1, J - 1)
40  STOP
50  NEXT J
60  NEXT I
] RUN

BREAK    IN  40
] ? I, J, A (I, J)
```

```
1                    1                    2
] CONT
BREAK IN 40
] ? I, J, A (I, J)
1                    2                    1
] CONT
BREAK   IN   40
] ? I, J, A (I, J)
2                    1                    2
] CONT
BREAK   IN   40
] ? I, J, A (I, J)
2                    2                    3
] CONT
```

4. 采用 DSP 语句

DSP 是一条 INTBASIC 语言的语句，可以采用立即和延迟两种执行方式。执行此语句后，屏幕能显示出 DSP 后面变量每次变化后的值，及变量所在的程序行行号。在采用立即执行方式时，不能在执行 DSP 命令后用 RUN 命令，因为 RUN 命令会使 DSP 命令无效，这时应采用 CON 或 GOTO 命令来继续执行程序。若要解除 DSP 方式（显示修改后的变量值方式）可执行 NODSP 命令。举例如下:

```
>LIST
     1    DSP   I
     2    DSP   J
     3    DSP   N
     5    DSP   M
     10   FOR   I=1   TO 2
     20   FOR   J=1   TO 2
     30   N=N+1
     40   M=M+N+I+J
     50   NEXT   J
     60   NEXT   I
     70   END
>RUN
#10   I=1
#20   J=1
#30   N=1
#40   M=3
#50   J=2 #30 N=2
```

```
#40  M=8
#50  J=3
#60  I=2 #20 J=1
#30  N=3
#40  M=14
#50  J=2 #30 N=4
#40  M=22
#50  J=3
#60  I=3

>LIST
   10  FOR I=1 TO 2
   20  FOR J=1 TO 2
   30  N=N+1
   40  M=M+N+I+J
   50  NEXT J
   60  NEXT I
   70  END

>DSP    M
>RUN
```

使用 RUN 命令使 DSP 失效，应改用 GOTO 命令:

```
>GOTO10
#40  M=71
#40  M=84
#40  M=98
#40  M=114

>NODSP
```

5. 采用 PRINT 语句

在程序的一些需观察变量变化情况的关键地方，可以暂时放置一些额外的 PRINT 语句，以便观察变量的变化情况。采用这种方式可以跟踪执行的流程和检查程序运行中变量变化的中间过程。举例如下:

```
5   A (0, 0) =1: A (0, 1) =1
10  FOR I =1 TO 2
20  FOR J =1 TO 2
30  A (I, J) = A (I-1, J) +A (I-1, J-1)
40  PRINT " I= " ; I , " J="; J , " A ( " ; I ; " , " ; J ;
```

```
") =" ; A (I, J)
50  NEXT J
60  NEXT I
] RUN
I=1            J=1            A (1, 1) =2
I=1            J=2            A (1, 2) =1
I=2            J=1            A (2, 1) =2
I=2            J=2            A (2, 2) =3

] 40

] LIST

5   A (0, 0) =1: A (0, 1) =1
10  FOR I=1 TO 2
20  FOR J=1 TO 2
30  A (I, J) =A (I-1, J) +A (I-1, J-1)
40  NEXT J
50  NEXT I
```

24. 中华学习机 ASCII 码的含义是什么

由于计算机只能处理二进制的信息，不能直接处理英文大写和小写字母、运算符号。以及其它专用符号。为了解决这个问题，在计算机中采用了一种用二进制数码来表示字母、符号和数字处理方法。这种处理方法采用的编码通常是美国信息交换标准代码 ASCII（ AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE)。

在中华学习机中采用的 ASCII 码有三种，一种是机内正 ASCII 码，一种是机内负 ASCII 码，另外一种是显示用的 ASCII 码。这三种 ASCII 码有着不同的含义，相互间存在一定的关系。

FPBASIC 程序存入主机内存和磁盘中，是采用正 ASCII 码，其代码有 128 个（$ 00-$ 7F)，见表 24-1 和表 24-2，例如字母 A 的 ASCII 码是 $ 41，数字 2 的 ASCII 码是 $ 32。由 $ 80 开始（大于 $ 7F）的 107 个 ASCII 码，用来代表 FPBASIC 中的保留字（见问题"FPBASIC 语言保留字的代码是什么"）。大于 $ EA 的几个 ASCII 码用于 FPBASIC 的错误信息。

负 ASCII 码实际是将相应的正 ASCII 码加 128（即 $ 80)，也就是将八位二进制数的最高位置 1，因此其代码的取值范围为 $ 80～ $ FF。例如，字母 A 的 ASCII 码为 $ C1，数字 2 的 ASCII 码为 $ B2。负 ASCII 码用于 INTBASIC 程序。键盘输入时，送至键盘

接收内存单元（$ C000）的 ASCII 码是负 ASCII 码。INTBASIC 语言中的保留字和一些符号的代码占用小于 $ 80 的数值范围（见问题“INTBASIC 语言保留字和符号的代码是什么”）。

另一种 ASCII 码用于显示屏显示字符用，叫显示 ASCII 码。在文本显示内存区域的内存单元中存入显示 ASCII 码，就可以在屏幕相应位置显示出 ASCII 码对应的字符。例如，在 $ 430 中存入数值 170 就会在屏幕第九行处显示一个“ * ”。显示 ASCII 码及对应字的字符见表 24–3。

表 24–1　键符对应的正 ASCII 码

键　符	Normal	Shift	Control	Both
) 0	$ 30 (48)	$ 29 (41)	$ 30 (48)	$ 29 (41)
! 1	$ 31 (49)	$ 21 (33)	$ 31 (49)	$ 21 (33)
@ 2	$ 32 (50)	$ 40 (64)	$ 32 (50)	$ 40 (64)
# 3	$ 33 (51)	$ 23 (35)	$ 33 (51)	$ 23 (35)
$ 4	$ 34 (52)	$ 24 (36)	$ 34 (52)	$ 24 (36)
% 5	$ 35 (53)	$ 25 (37)	$ 35 (53)	$ 25 (37)
ˆ 6	$ 36 (54)	$ 5E (94)	$ 1E (30)	$ 1E (30)
& 7	$ 37 (55)	$ 26 (38)	$ 37 (55)	$ 26 (38)
* 8	$ 38 (56)	$ 2A (42)	$ 38 (56)	$ 2A (42)
(9	$ 39 (57)	$ 28 (40)	$ 39 (57)	$ 28 (40)
_ –	$ 2D (45)	$ 5F (95)	$ 2D (45)	$ 1F (31)
+ =	$ 3D (61)	$ 2B (43)	$ 3D (61)	$ 2B (43)
| \	$ 5C (92)	$ 7C (124)	$ 1C (28)	$ 1C (28)
{ [	$ 5B (91)	$ 7B (123)	$ 1B (27)	$ 1B (27)
}]	$ 5D (93)	$ 7D (125)	$ 1D (29)	$ 1D (29)
: ;	$ 3B (59)	$ 3A (58)	$ 3B (59)	$ 3A (58)
” ’	$ 27 (39)	$ 22 (34)	$ 27 (39)	$ 22 (34)
~ `	$ 60 (96)	$ 7E (126)	$ 60 (96)	$ 7E (126)
< ,	$ 2C (44)	$ 3C (60)	$ 2C (44)	$ 3C (60)
> .	$ 2E (46)	$ 3E (62)	$ 2E (46)	$ 3E (42)
? /	$ 2F (47)	$ 3F (63)	$ 2F (47)	$ 3F (63)

续表 24-1

Test	$ 16 (22)	$ 16 (22)	$ 16 (22)	$ 16 (22)
A	$ 41 (65)	$ 41 (65)	$ 01 (01)	$ 01 (01)
B	$ 42 (66)	$ 42 (66)	$ 02 (02)	$ 02 (02)
C	$ 43 (67)	$ 43 (67)	$ 03 (03)	$ 03 (03)
D	$ 44 (68)	$ 44 (68)	$ 04 (04)	$ 04 (04)
E	$ 45 (69)	$ 45 (69)	$ 05 (05)	$ 05 (05)
F	$ 46 (70)	$ 46 (70)	$ 06 (06)	$ 06 (06)
G	$ 47 (71)	$ 47 (71)	$ 07 (07)	$ 07 (07)
H	$ 48 (72)	$ 48 (72)	$ 08 (08)	$ 08 (08)
I	$ 49 (73)	$ 49 (73)	$ 09 (09)	$ 09 (09)
J	$ 4A (74)	$ 4A (74)	$ 0A (10)	$ 0A (10)
K	$ AB (75)	$ 4B (75)	$ 0B (11)	$ 0B (11)
L	$ 4C (76)	$ 4C (76)	$ 0C (12)	$ 0C (12)
M	$ 4D (77)	$ 4D (77)	$ 0D (13)	$ 0D (13)
N	$ 4E (78)	$ 4E (78)	$ 0E (14)	$ 0E (14)
O	$ 4F (79)	$ 4F (79)	$ 0F (15)	$ 0F (15)
P	$ 50 (80)	$ 50 (80)	$ 10 (16)	$ 10 (16)
Q	$ 51 (81)	$ 51 (81)	$ 11 (17)	$ 11 (17)
R	$ 52 (82)	$ 52 (82)	$ 12 (18)	$ 12 (18)
S	$ 53 (83)	$ 53 (83)	$ 13 (19)	$ 13 (19)
T	$ 54 (84)	$ 54 (84)	$ 14 (20)	$ 14 (20)
U	$ 55 (85)	$ 55 (85)	$ 15 (21)	$ 15 (21)
V	$ 56 (86)	$ 56 (86)	$ 16 (22)	$ 16 (22)
W	$ 57 (87)	$ 57 (87)	$ 17 (23)	$ 17 (23)
X	$ 58 (88)	$ 58 (88)	$ 18 (24)	$ 18 (24)
Y	$ 59 (89)	$ 59 (89)	$ 19 (25)	$ 19 (25)
Z	$ 5A (90)	$ 5A (90)	$ 1A (26)	$ 1A (26)

续表 24-1

键 符	Normal	Shift	Control	Both
中 文	$1F (31)	$1F (31)	$1F (31)	$1F (31)
西 文	$11 (17)	$11 (17)	$11 (17)	$11 (17)
F1	$01 (01)	$01 (01)	$01 (01)	$01 (01)
F2	$0C (12)	$0C (12)	$0C (12)	$0C (12)
F3	$17 (23)	$17 (23)	$17 (23)	$17 (23)
F4	$14 (20)	$14 (20)	$14 (20)	$14 (20)
F5	$06 (06)	$06 (06)	$06 (06)	$06 (06)
ESC	$1B (27)	$1B (27)	$1B (27)	$1B (27)
TAB	$09 (09)	$09 (09)	$09 (09)	$09 (09)
Quit	$03 (03)	$03 (03)	$03 (03)	$03 (03)
Return	$0D (13)	$0D (13)	$0D (13)	$0D (13)
空格	$20 (32)	$20 (32)	$20 (32)	$20 (32)
◁	$08 (08)	$08 (08)	$08 (08)	$08 (08)
△	$0B (11)	$0B (11)	$0B (11)	$0B (11)
▽	$0A (10)	$0A (10)	$0A (10)	$0A (10)
▷	$15 (21)	$15 (21)	$15 (21)	$15 (21)

注：(1) 键盘接收单元（$C000）的 ASCII 码是高位为 1 的负 ASCII 码，高位为 1 表示按了某个按键，低七位存放的是相应的正 ASCII 码，表 24-1 就是这种正 ASCII 码表。

(2) Normal 栏表示只按某字符键或功能键时的正 ASCII 码。Shift 栏表示按某键的同时也按 SHIFT 键产生的正 ASCII 码。Control 一栏表示同时按某键和 CONT 键所产生的正 ASCII 码。both 一栏表示同时按 CONT、SHIFT 和某键所产生的正 ASCII 码。

表 24-2 正 ASCII 码

正 ASCII 码	显示字符	键	正 ASCII 码	显示字符	键
0		CTRL-@	22		CTRL-V
1		CTRL-A	23		CTRL-W
2		CTRL-B	24		CTRL-X
3		CTRL-C	25		CTRL-Y
4		CTRL-D	26		CTRL-Z
5		CTRL-E	27		ESC
6		CTRL-F	28		n/a

续表 24-2

正 ASCII 码	显示字符	键	正 ASCII 码	显示字符	键
7	(响一声)	CTRL-G	29		CTRL-SHIFT-M
8	光标左移一位	CTRL-H 或←	30		CTRL-∧
			31	@	n/a
9		CTRL-I	32	space	space（空格）
10		CTRL-J	33	!	!
11		CTRL-K	34	"	"
12		CTRL-L	35	#	#
13		CTRL-M	36	$	$
		Return	37	%	%
14		CTRL-N	38	&	&
15		CTRL-O	39	'	'
16		CTRL-P	40	(	(
17		CTRL-Q	41	)	)
18		CTRL-R	42	*	*
19		CTRL-S	43	+	+
20		CTRL-T	44	,	,
21	光标右移一位	CTRL-U 或→	45	–	–
			46	.	.
47	/	/	72	H	H
48	0	0	73	I	I
49	1	1	74	J	J
50	2	2	75	K	K
51	3	3	76	L	L
52	4	4	77	M	M
53	5	5	78	N	N
54	6	6	79	O	O
55	7	7	80	P	P
56	8	8	81	Q	Q
57	9	9	82	R	R
58	:	:	83	S	S
59	;	;	84	T	T
60	<	<	85	U	U
61	=	=	86	V	V
62	>	>	87	W	W
63	?	?	88	X	X

续表 24-2

正 ASCII 码	显示字符	键	正 ASCII 码	显示字符	键
64	@	@	89	Y	Y
65	A	A	90	Z	Z
66	B	B	91	[	[
67	C	C	92	＼	＼
68	D	D	93	]	]
69	E	E	94	∧	∧
70	F	F	95	—	n/a
71	G	G	96	`	`
97	a		113	q	
98	b		114	r	
99	c		115	s	
100	d		116	t	
101	e		117	u	
102	f		118	v	
103	g		119	w	
104	h		120	x	
105	i		121	y	
106	j		122	g	
107	k		123	{	
108	l		124	|	
109	m		125	}	
110	n		126	~	
111	o		127	■	
112	p				

表 24-3 显示 ASCII 码及其字符

		反转显示				闪烁显示				正常显示	
十进制 →	十六进制 →	0	16	32	48	64	80	96	112	128	144
↓	↓	$ 00	$ 10	$ 20	$ 30	$ 40	$ 50	$ 60	$ 70	$ 80	$ 90
0	$ 0	(a)	P		0	@	P	0		@	P
1	$ 1	A	Q	!	1	A	Q	!	1	A	Q
2	$ 2	B	R	″	2	B	R	″	2	B	R
3	$ 3	C	S	#	3	C	S	#	3	C	S
4	$ 4	D	T	$	4	D	T	$	4	D	T
5	$ 5	E	U	%	5	E	U	%	5	E	U
6	$ 6	F	V	&	6	F	V	&	6	F	V
7	$ 7	G	W	′	7	G	W	′	7	G	W
8	$ 8	H	X	(	8	H	X	(	8	H	X
9	$ 9	I	Y	)	9	I	Y	)	9	I	Y
10	$ A	J	Z	*	:	J	Z	*	:	J	Z
11	$ B	K	[	+	;	K	[	+	;	K	[
12	$ C	L	＼	,	<	L	＼	,	<	L	＼
13	$ D	M	]	—	=	M	]	—	=	M	]
14	$ E	N	^	.	>	N	^	.	>	N	^
15	$ F	O	—	/	?	O	—	/	?	O	

十进制→ 十六进制→ ↓ ↓		正常显示				小写正常显示	
		160	176	192	208	224	240
		$A0	$B0	$C0	$D0	$E0	$F0
0	$0		0	@	P		p
1	$1	!	1	A	Q	a	q
2	$2	″	2	B	R	b	r
3	$3	#	3	C	S	c	s
4	$4	$	4	D	T	d	t
5	$5	%	5	E	U	e	u
6	$6	&	6	F	V	f	v
7	$7	′	7	G	W	g	w
8	$8	(	8	H	X	h	x
9	$9	)	9	I	Y	i	y
10	$A	*	:	J	Z	j	z
11	$B	+		K	[	k	{
12	$C	,	<	L	/	l	/
13	$D	−	=	M	]	m	}
14	$E	.	>	N	ˆ	n	~
15	$F	/	?	O	−	o	■

25. 如何检验中华学习机的 ASCII 码与字符的对应关系

编写几个简单的 BASIC 程序，然后运行它们即可检验中华学习机的 ASCII 码与字符的对应关系。

1. 检验机内正 ASCII 码与字符的对应关系运行下面的程序即可:

```
5    REM PROGRAM 25.1
10   FOR I = 0 TO 127
20   PRINT I ; “ : ” ; CHR$ ( I ),
40   NEXT I
```

2. 检验机内显示 ASCII 码与字符的对应关系运行下面的程序即可:

```
1    REM PROGRAM 25.2
5    HOME
10   FOR I = 1 TO 55
20   VTAB 11: HTAB 18: PRINT I ; ″ : ″ : POKE 1342, I
30   FOR J = 1 TO 200: NEXT J
40   NEXT I
```

3. 检验键盘输入的 ASCII 码与字符的对应关系

这是检验按下键盘某个按键后，键盘接收内存单元中的 ASCII 码（负 ASCII 码）与按下键的对应关系。运行下面这个程序，然后按相应的键即可。

```
1   REM PROGRAM 25.3
5   HOME
10  P = PEEK ( 49152 )
20  IF P < 128 THEN 10
30  POKE－16368, 0
40  PRINT P ;  " :  " ;  CHR$  ( P－128 ),
50  GOTO 10
```

26. FPBASIC 语言保留字的代码是什么

FPBASIC 语言保留字及其代码按英文字母顺序给出的见表 26-1，按代码数值顺序给出的见表 26-2。

表 26-1 FPBASIC 语言保留字及代码

保留字	代码		保留字	代码	
	十进制	十六进制		十进制	十六进制
ABS	(212)	D4	NORMAL	(157)	9D
AND	(205)	CD	NOT	(198)	C6
ASC	(230)	E6	NOTRACE	(156)	9C
AT	(197)	C5	ON	(180)	B4
ATN	(225)	E1	ONERR	(165)	A5
CALL	(140)	8C	OR	(206)	CE
CHR$	(231)	E7	PDL	(216)	D8
CLEAR	(189)	BD	PEEK	(226)	E2
COLOR=	(160)	A0	PLAY	(236)	ED
CONT	(187)	BB	PLOT	(141)	8D
COS	(222)	DE	POKE	(185)	B9
DATA	(131)	83	POP	(161)	A1
DEF	(184)	B8	POS	(217)	D9
DEL	(133)	85	PRINT	(186)	BA
DIM	(134)	86	PR#	(138)	8A
END	(128)	80	READ	(135)	87
EXP	(221)	DD	RECALL	(167)	A7

续表 26-1

FLASH	(159)	9F	REM	(178)	B2
FN	(194)	C2	RESTORE	(174)	AE
FOR	(129)	81	RESUME	(166)	A6
FRE	(214)	D6	RETURN	(177)	B1
GET	(190)	BE	RIGHT $	(233)	E9
GOSUB	(176)	B0	RND	(219)	DB
GOTO	(171)	AB	ROT=	(152)	98
GR	(136)	88	RUN	(172)	AC
HCOLOR=	(146)	92	SAVE	(183)	B7
HGR	(145)	91	SCALE=	(153)	99
HGR2	(144)	90	SCRN (	(215)	D7
HIMEM#	(163)	A3	SGN	(210)	D2
HLIN	(142)	8E	SHLOAD	(154)	9A
HOME	(151)	97	SIN	(223)	DF
HPLOT	(147)	93	SPC (	(195)	C3
HTAB	(150)	96	SPEED=	(169)	A9
IF	(173)	AD	SQR	(218)	DA
IN#	(139)	8B	STEP	(199)	C7
INPUT	(132)	84	STOP	(179)	B3
INT	(211)	D3	STORE	(168)	A8
INVERSE	(158)	9E	STR $	(228)	E4
LEFT $	(232)	E8	TAB (	(192)	BF
LEN	(227)	E3	TAN	(224)	E0
INT	(170)	AA	TEXT	(137)	89
LG	(237)	EE	THEN	(196)	C4
LIST	(188)	BC	TO	(193)	C0
LOAD	(182)	B6	TRACE	(155)	D5
LOG	(220)	DC	USR	(213)	EA
LOMEM:	(164)	A4	VAL	(229)	8F
MID $	(234)	EA	VLIN	(143)	A2
MUSIC	(235)	EB	VTAB	(162)	B5
NEW	(191)	BF	WAIT	(181)	95
NEXT	(130)	82	XDRAW	(149)	

27、INTBASIC 语言保留字和符号的代码是什么

INTBASIC 语言的保留字见表 27–1。

表 27–1 整数 BASIC 保留字

ABS	END	LET	PDL	SAVE
AND	FOR	LIST	PEEK	SCRN
ASC	GOSUB	LOAD	PLOT	SGN
AT	GOTO	LOMEM	POKE	STEP
AUTO	GR	MAN	POP	TAB
CALL	HIMEM	MOD	PRINT	TEXT
COLOR=	HLIN	NEW	PR#	THEN
CON	IF	NEXT	REM	TO
DEL	IN#	NOT	RETURN	TRACE
DIM	INPUT	NOTRACE	RND	VLIN
DSP	LEN	OR	RUN	VTAB

INTBASIC 语言的保留字和符号的代码见表 27–2。

表 27–2 INTBASIC 语言保留字及符号的代码

	0	1	2	3	4	5	6	7
0	HIMEM	HIMEM	ˆ	SGN	$ 字符串变量名尾符	TAB	IF	= 串型变量赋值号
1	程序行结束符	LOMEM	+	ABS	$	END	PRINT 后跟串型变量	= 数值变量赋值号
2	–	+	(用在对字符定义长度的 DIM 中	PDL	(用于字符串某一字符位置取得字符	INPUT 后跟串型量	PRINT 后跟数型量	) 合法的右括号
3	: 程序行中两个语句分隔符	– 减	, DIM 中两个最大下标值中间	RNDX	, 用在 DIM 中，其后说明的是串变量	INPUT 其后带提示字符串	PRINT 其后无输出项	)

续表 27-2

	0	1	2	3	4	5	6	7
4	LOAD	* 乘	THEN 后跟行号	(DIM 说明的第一个名字为数组	% 用 DIM 对多个名字说明时%其后为数组	INPUT 后跟数型变量或数组元素	POKE	LIST 后跟行号
5	SAVE	/ 除	THEN 后跟行号	+ 正	; 用在 PRINT 中%其后为串型量	FOR	, 跟在 POKE 之后	, 跟LIST 之后
6	CON	= 相等对数型量	, 用于 INPUT 中，其后为字符串变量名	− 负	; 用在 PRINT 中%其后为数型量	= 循环变量赋初值	COLOR =	LIST 列示全部程序
7	RUN 带行号	不等对的数型量	, 用于 INPUT 中，其后为数型量	NOT	; 用在 PRINT 语句尾	TO	PLOT	POP
8	RUN 不带行号	>=	” 字符串常量左引号	(表达式左括号	, 用在 PRINT 中其后为串型量	STEP	, 用在 PLOT 中	NODSP 对串型变量
9	DEL	>	” 字符串常量右引号	= 相等对串型量	, 用在 PRINT 中其后为数型量	NEXT	HLIN	NODSP 对数型量
A	, 用在 DEL 命令中	<=	(用于使用串型量中分量	# 不等对串型量	, 用于 PRINT 语句尾	, 用于对多个循环变量 NEXT 时的分隔	, 用在 HLIN 中	NOTRACE
B	NEW	< >	!	LEN(	TEXT	RETURN	AT	DSP 后跟串变量
C	CLR	<	!	ASC(	GR	GOSUB	VLIN	DSP 后跟对数型量

续表 27–2

	0	1	2	3	4	5	6	7
D	AUTO	AND	(对数组元素使用或跟 TAB 之后	SCRN(	CALL	REM	, 用在 VLIN 中	TRACE
E	, 用在 AUTO 命令中	OR	PEEK	, 用于 SCRN 函数中	DIM 对字符说明	LET	AT	PR#
F	MAN	MOD	RND	(函数左括号	DIM 对数组说明	GOTO	VTAB	IN#

28、FPBASIC 语言错误信息的含义是什么

FPBASIC 语言错误信息的含义如下:

? BAD SUBSCRIPT ERROR

数组下标超界或个数不符。错误代码是 107。

? CAN'T CONTINUE ERROR

根本没有程序，或者是已经严重错误出现，或者已经修改过程序之后，试图使用 CONT 命令继续程序的运行。

? DIVISION BY ZERO ERROR

除数为零，错误代码是 133。

? FORMULA TOO COMPLEX ERROR

如果IF string THEN 这样的语句已经执行了两次以上的话，就会产生这一信息。错误代码为 191。

? ILLEGAL DIRECT ERROR

在直接式中执行INPUT、DEF FN或GET命令。

? ILLEGAL QUANTITY ERROR

在一个字符串函数、数值函数、画图语句等中，所用到的数值超出了能够接受的范围。错误代码为 53。

? NEXT WITHOUT FOR ERROR

执行到一个没有 FOR 对应的 NEXT 语句。错误代码为 0。

? OUT OF MEMORY ERROR

内存不够,可能是下列情况引起: 程序太长，变量太多，FOR 循环多于十层，子程序超过 24 层，括号超过 36 层，LOMEM: 定得太高，HIMEM: 定得太低错误代码为 77。

? OUT OF DATA ERROR

在 DATA 的资料不够 READ 读取。错误代码为 42。

? OVERFLOW ERROE

输入或程序计算出来的数字太大或太小了；计算机能被接受的数字范围在−1.7E+38到 1.7E+38 之间。错误代码为 69。

? REDIM'D ARRAY ERROR

数组定维重复。错误代码为 12。

? RETURN WITHOUT GOSUB ERROR

执行到一个没有 GOSUB 与之相对应的 RETURN 语句。错误代码为 22。

? STRING TOO LONG ERROE

把各个字符串连接的时候，所得出来的字符串的长度超过255。错误代码为176。

? SYNTAX ERROR

拼写、标点或次序等错误或者是任何不能由其它信息表示的错误。错误代码为16。

? TYPE MISMATCH ERROR

变量类型不符，错误代码为 163。

? UNDEF'D FUNCTION ERROR

使用一个没有定义的函数，错误代码为 244。

? UNDEF'D STATEMENT ERROR

程序企图转到一个不存在的程序行。错误代码为 90。

FPBASIC 出错信息的编号，存放在地址为:
$DE(222)的内存单元中。

29、INTBASIC 语言错误信息的含义是什么

INTBASIC 语言错误信息的含义如下

* * * >255 ERR

一个应该在 0～255 之间的值超出了范围。

* * * >32767 ERR

已经算出或输入了某个大于 32767 或小于−32767 的数值。

* * * 16 FORS ERR

已经有 16 个 FOR 同时在执行。

* * * 16 GOSUB ERR

已经有 17 个 GOSUB 在执行了，但从第一个到现在还没有执行任何 RETURN 语句。

* * * DIM ERR

同样名称的数组被重复定维。

* * * BAD RETURN ERR

执行到一不与任何 GOSUB 对应的 RETURN 语句。

* * * MEM FULL ERR

内存已经不够用了。

* * * NO END ERR

程序中最后一句被执行语句不是 END。

* * * RANGE ERR

这种信息表示两种可能的情况：一是数组元素的下标值小于零或大于数组容量；二是 KLIN、VLIN、PLOT、TAB 或 VTAB 语句中的值超出了即有的范围。

* * * RETYPE LINE

由INPUT所产生的错误，这会先产生一道错误，然后才是这道信息，请你把那一列重新打过。

* * * STR ERR

企图执行一个不合法的字符串运算。

* * * STR OVEL ERR

一个字符串已存入比你的容量还多的字符了。

* * * SYNTAX ERR

这是指拼写，标点，有关书写次序，以及不能由其它错误信息所表示的信息。

* * * TOO LONG ERR

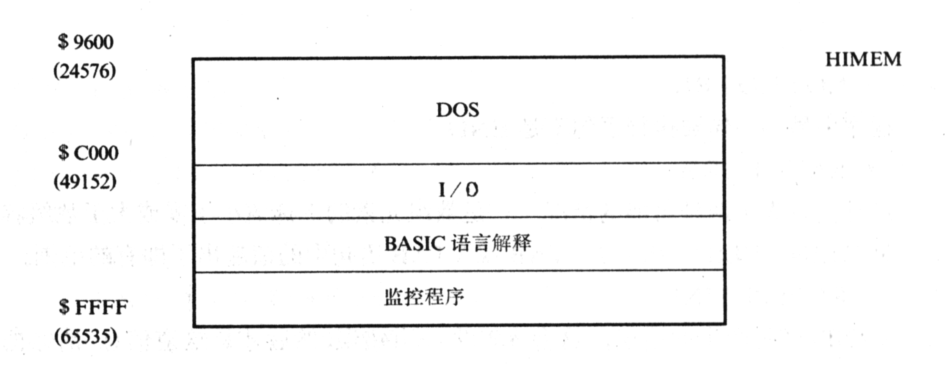
一行已经超过了 128 个文字，或者有已经超过了 12 层括号了。

30、FPBASIC 程序与 INTBASIC 程序在内存中存放的特点有何不同

用户程序、变量和字符串数据在内存中存放的位置是依据 BASIC 语言类型而确定的。对于 FPBASIC 语言来说程序在内存中是从低地址内存单元（$ 800），向高地址内存单元存放。对于 INTBASIC 语言来说，整个程序在内存中是从高地址内存单元（引导 DOS 后是 $ 95FF）向低地址内存单元存放。对于整个程序中各个程序行来说，两种类型的 BASIC 程序者是依程序行号由小至大的次序，从低地址内存单元向高地址内存单元存放。存放情况可参看表 30–1。

表 30–1　BASIC 程序存放特点

地址			
$ 000	系统区		
$ 0800	FPBASIC 程序	变量数组数据区	LOMEM
	简单变量	INTBASIC 程序	
	数组	↑ ↑	
	字符串 ↑		



31. FPBASIC　程序在内存程序区中是如何存放的

FPBASIC 程序在内存程序区中的存放格式是一种链式结构，其特点是找到一行程序，就可以知道下一行程序的位置。存放格式如表 31−1 所示。

表 31−1　FPBASIC 程序在内存中的存放格式

		$ 800
		0 0(FPBASIC 程序标志)
一个程序行	链指针	下一个程序行首地址低位
		下一个程序行首地址高位
		本程序行行号低位
		本程序行行号高位
		本程序行各语句
		0 0
一个程序行	链指针	下个程序行首地址低位
		下个程序行首地址高位
		本程序行行号低位
		本程序行行号高位
		本程序行各语句
		0 0
		: : :
一个程序行	链指针	下一个程序行首地址低位
		下一个程序行首地址高位
		本程序行首地址低位
		本程序行行号低位
		本程序行行号高位
		本程序各语句
		0 0
程序结尾标志		0 0
		0 0

下面举例进行说明:

程序是:

```
10   INPUT N,M
20    $ ="ABCDEFGHIJKLM"
30   B $ ="NOPQRSTUVWXYZ"
40   PRINT A,B,A $ ,B $
50   FIR I = 1 TO 10
60   READ N
70   M = M+N
80   NEXT I
90   PRINT "M =";M
100   DATA 1,2,3,4,5,6,7,8,9,0
```

内存程序区中 ASCII 码数据:

```
]CALL−151
* 800.898
0800− 00 0A 08 0A 00 84 4E 2C
0808− 4D 00 21 08 14 00 41 24
0810− D0 22 41 42 43 44 45 46
0818− 47 48 49 4A 4B 4C 4D 22
0820− 00 38 08 1E 00 42 24 DO
0828− 22 4E 4F 50 51 52 53 54
0830− 55 56 57 58 59 5A 22 00
0838− 47 08 28 00 BA 41 2C 42
0840− 2C 41 24 2C 42 24 00 53
0848− 08 32 00 81 49 D0 31 C1
0850− 31 30 00 5A 08 3C 00 87
0858− 4E 00 64 08 46 00 4D D0
0860− 4D CB 4E 00 6B 08 50 00
0868− 82 49 00 77 08 5A 00 BA
0870− 22 4D 3D 22 3B 4D 00 91
0878− 08 64 00 83 20 31 2C 32
0880− 2C 33 2C 34 2C 35 2C 36
0888− 2C 37 2C 38 2C 39 2C 30
0890− 00 00 00 0A 00 00 00 0A
0898− 2D
```

以第一程序行为例分析如下

$ 801、$ 802 内存单元中存放下一程序行的首地址 $ 080A, $ 0803, $ 0804 存放行号 10, $ 805 中的数据 $ 84 是 INPUT 字保留的 ASCII 码 $ 806~ $ 808 中的数据 $ 4E, $ 2C, $ 4D 分别是字符 N、，、M 的 ASCII 码, $ 809 中数据 $ 00 是本程序行的结束符。$ 80A 是下一程序行的首地址。

由上面数据可以看出, 本程序占用了 $ 800~$892 内存区域, $ 890~ $ 892 内存单元中为三个连续的零。

32. INTBASIC 程序在内存程序区中是如何存放的

INTBASIC 程序总是紧邻着 HIMEM：指向的内存单元（机内有 DOS 情况下为 $ 9600）存放的。行号小的程序存放的内存单元地址号码小（离 HIMEM 远），行号大的程序存放的内存单元地址号码大（离 HIMEM 近），按行号大小顺序存放在内存单元中。

对于程序中每一个程序行，它的第一个内存单元存放这个程序行的长度（所用内存单元个数）。第二、三个内存单元存放程序行的行号。从第四个内存单元开始存放程序行中各语句。每一个程序行的结尾用 $ 01 数作为标志（它实际是回车键转化来的）。

INTBASIC 程序存放的结构特点可参看表 32-1。这里需注意的是：存放在内存中 INTBASIC 程序的变量名所用的字符和字符串中的字符采用负 ASCII 码，其值均大于或等于 $ 80，而保留字及部分符号（运算符等）的代码取值范围为 $ 00～ $ FF。这与 FPBASIC 程序不一样。有关内容可参看问题 24 与问题 27。

表 32-1　INTBASIC 在内存中的存放格式

地址	内容
程序区首（$ CA、 $ CB）	第一个程序行长度
	第一个程序行行号低八位
	第一个程序行行号高八位
	语　　句
	$ 01
	第二个程序行长度
	第二个程序行行号低八位
	第二个程序行行号高八位
	语　　句
	$ 01
	⋮
	最后一个程序行长度
	最后一个程序行行号低八位
	最后一个程序行行号高八位
（$ 4C、 $ 4D）	语　　句
HIMEM： $ 9600	$ 01

举例如下：

```
>LIST
      10  DIM A$ (13) , B $ (15) , A (4)
      20  A$ = "ABCDEFGHIJKLM"
      30  B$ = "NOPQRSTUVWXYZ"
      40  FOR I=1 TO 4
      50  LET A ( I ) =I
```

```
  60   NEXT I
  70   PRINT A $, B $
  80   END
>RUN
ABCDEFGHIJKLM    NOPQRSTUVWXYZ
>CALL-151
* 4A . 4D
004A- 00 08 00 96
* CA. CD
00CA- 8B 95 3C 08
*958B. 9600
958B- 1B 0A 00 4E C1
9590- 40 22 B1 0D 00 72 43 C2
9598- 40 22 B1 0F 00 72 44 C1
95A0- 34 B4 04 00 72 01 16 14
95A8- 00 C1 40 70 28 C1 C2 C3
95B0- C4 C5 C6 C7 C8 C9 CA CB
95B8- CC CD 29 01 16 1E 00 C2
95C0- 40 70 28 CE CF D0 D1 D2
95C8- D3 D4 D5 D6 D7 D8 D9 DA
95D0- 29 01 0E 28 00 55 C9 56
95D8- B1 01 00 57 B4 04 00 01
95E0- 0B 32 00 5E C1 2D C9 72
95E8- 71 C9 01 06 3C 00 59 C9
95F0- 01 0A 46 00 61 C1 40 48
95F8- C2 40 01 05 50 00 51 01
9600- 62
```

以第一个程序行为例，该程序行共占用了＄958B～＄95A5 共 27 个内存单元，故第一个内存单元存放程序行长度＄1B。第二、三个内存单元存放行号＄0A（即 10）。以后的内存单元存放语句的 ASCII 码或保留字代码及结束符＄01。其中＄4E 是保留字 DIM 代码（对定符进行说明的）。＄C1 是字符 A 的负 ASCII 码，＄40 是字符串变量名尾符“S”的代码，＄22 是左括号“(”的代码，其后是字符串长度值，＄9593、＄9594 中存放的＄0D、＄00 是数值 13，＄9592 中存放的＄B1 是换码后留下的（键盘输入 1 的负 ASCII 码为＄B1），＄72 为右括号“)”的代码，＄43 是“,”的代码，并说明其后是字符串变量，＄C2 是字符 B 的负 ASCII 码，＄44 也是“,”的代码，但说明其后是数组，＄34 是“(”的代码，其后是数组的最大下标值。

33. 与 FPBASIC 程序有关的向量指针主要有哪些

在 FPBASIC 程序输入、调试、运行和存入磁盘等过程中，需要借助内存 0 页中一些向量指针来完成工作。主要的向量指针有以下几个：

1. ＄67、＄68：存放 FPBASIC 源程序的起始地址，其初始值为＄0801，它在工作过程中始终保持不变（不使用 LOMEM 命令的情况下）。由于用户区起始地址需要存放一个字节的 BASIC 类型标志，故 FPBASIC 源程序起始地址为用户起始地址加 1。

2. ＄69、＄6A：存放 FPBASIC 变量表首地址，其初始值为＄0803。变量表紧跟在源程序的尾部（连续三个 0 之后），源程序长度变化后，该地址也随之变化。

3. $ 6B、$ 6C：存放 FPBASIC 数组表首地址，其初值为 $ 0803。数组表紧跟在变量表之后，如果没有变量表，则紧跟在源程序尾部。数组表首地址随源程序和变量表长度变化而改变。

4. $ 6D、$ 6E：存放数组表尾地址，其初始值为 $ 803。由于数组表的结构长度（即占内存量）是在使用 DIM 语句后确定下来的，故不但需要指示首地址，还应指示尾地址。数组表尾地址会随数组表首地址和数组表内容的变化而改变。

5. $ 6F、$ 70：存放字符串数据尾地址，其初始值为 $ 9600（在机内有 DOS 情况下）。这里所说的字符串数据是指程序运行中用 INPUT 或 GET 语句输入的字符串，以及程序运行中产生的新字符串。这些字符串数据在内存中存放的顺序与源程序和变量相反，它是由高地址端向低地址端存放。

6. $ 73、$ 74：存放用户可用的内存地址最大值（即 HIMEM 指示的值），其初始值为 $ 9600（在机内有 DOS 情况下）。

7. $ 75、$ 76：存放当前处理行号。它是 BASIC 解释程序的工作指针。如果是立即执行方式，则它们置为 $ FF。

8. $ 7B、$ 7C：存放 DATA 数据正在被 READ 语句读取的当前行号。

9. $ 7D、$ 7E：存放 DATA 数据正在被 READ 语句读取的绝对内存单元地址。

10. $ AF、$ B0：存放源程序的尾地址，其初始值为 $ 0803。该值随源程序长度的变化而改变。改变此数据会对后面的变量表和数组表数据做整体移动。源程序尾地址是指源程序结尾三个 0 后面那个内存单元地址（在机内有 DOS 时，它指三个 0 后面第二个内存单元地址）。

11. $ DA、$ DB：存放发生错误的语句行行号。

12. $ DE：存放错误代码。

以上指针归纳如表 33–1 所示：

表 33–1 FPBASIC 向量指针

向量指针作用	指针地址		初始值
	十六进制	十进制	
FPBASIC 源程序首址	$ 67	103	$ 01
	$ 68	104	$ 08
变量表首址	$ 69	105	$ 03
	$ 6A	106	$ 08
数组表首址	$ 6B	107	$ 03
	$ 6C	108	$ 08
数组表尾址	$ 6D	109	$ 03
	$ 6E	110	$ 08
字符串数据尾址	$ 6F	111	$ 00
	$ 70	112	$ 96
用户最高可用地址 (HIMEM)	$ 73	115	$ 00
	$ 74	116	$ 96

续表 33-1

当前处理行号	$ 75	117	——
	$ 76	118	
正在读取的数据 DATA 语句行号	$ 7B	123	——
	$ 7C	124	
正在读取的数据的内存单元地址	$ 7D	125	$ 00
	$ 7E	126	$ 08
FPBASIC 源程序尾址	$ AF	175	$ 03
	$ B0	176	$ 08
出错语句的行号	$ DA	218	$ FF
	$ DB	219	$ FF
错 误 代 码	$ DE	220	$ FF

34. 与 INTBASIC 程序有关的向量指针主要有哪些

INTBASIC 程序存放在高地址内存单元中，而相应的变量数据存放在低地址内存单元中。在 0 页有八个内存单元，每两个一组。建立四个 INTBASIC 程序的向量指针。

$ CA. $ CB 两个内存单元存放 INTBASIC 程序区首地址，低八位在前，高八位在后，其初始值为 $ 9600。$ 4C、$ 4D 两个内存单元存放 INTBASIC 程序区原尾地址，其初始值为 $ 9600，而且在程序运行时它总是指向 $ 9600。

$ 4A. $ 4B 两个内存单元存放 INTBASIC 程序的变量（包含数组的下标变量）数据区的首地址，也是低八位在前，高八位在后，其初始值为 $ 800，而且其值在程序运行时不发生变化。$ CC、$ CD 两个内存单元存放 INTBASIC 程序变量数据区的尾地址，其初始值为 $ 800。

INTBASIC 程序的向量指针如表 34-1 所示。

表 34-1　INTBASIC 程序向量指针特点

向 量 指 针 名 称	指针（内存）地址	初 始 化 值
变量数据区首地址指针 (LOMEM:)	$ 4A	$ 00
	$ 4B	$ 08
INTBASIC 程序尾地址指针 (HIMEM:)	$ 4C	$ 00
	$ 4D	$ 96
INTBASIC 程序首地址指针	$ CA	$ 00
	$ CB	$ 96
变量数据区尾地址指针	$ CC	$ 00
	$ CD	$ 08

35. FPBASIC 简单变量表的结构特点是什么

简单变量分为整型、实型、字符串型和函数型变量四种，它是在程序运行过程中建立的。变量表中各变量占七个内存单元，前两个内存单元存放变量名的 ASCII 码，后面五个内存单元根据变量的不同类型存放相应形式的数值或地址。各变量表特点如下：

1. 整型变量表：在变量名后面第一个内存单元中存放整型数（十六位二进制数）的高八位，第二个内存单元存放整型数的低八位，其余三个内存单元存入“0”，如表 35–1(a)所示。因此整型数取值范围只能在 0～65535（即＄0000～＄FFFF）之间。

2. 实型变量表：它占满了七个内存单元。变量名后面用一个内存单元存放幂指数，其余四个内存单元存放尾数，如表 35–1（b）所示。

3. 字符串变量表：字符串变量名后面的内存单元存放字符串索引项。第一个内存单元存放字符串长度，因用一个内存单元存放字符串长度，所以要求字符串长度不得大于 255。后面两个内存单元存放字符串在内存中存放位置的首地址。最后两个内存单元存入“0”，如表 35–1（c）所示。字符串数据有两部分，一部分是源程序中由赋值、置数语句设定的，另一部分是程序运行后产生的。后一部分数据存放在用户区高地址内存单元中。

4. 函数型变量表：函数变量名后的两个内存单元依次填入函数公式（“＝”后边）的首地址的低位和高位 。其后两个内存单元存放函数自变量表数值的首地址，如果该变量初次使用，则函数型变量表之后建立一个该变量的实型变量表。最后一个内存单元存放公式中第一个字符的 ASCII 码，如表 35–1 所示。

上述四种类型的变量名，因类型不同而加入不同的标志。对于整型变量名，其存放变量名的内存单元最高位均置 1；对于实型变量名，其存放变量名的内存单元最高位均置 0；对于字符串型变量名，其存放变量名的第一个内存单元最高位置 0，第二个内存单元最高位置 1；对于函数型变量名，其存放变量名的第一个内存单元最高位置 1，第二个内存单元最高位置 0。以 A1 变量名为例，表 35–2 说明它们的区别。

表 35–1　四种类型的变量表

(a) 整型
变量名 1（负 ASCII 码）
变量名 2（负 ASCII 码）
数值的高八位
数值的低八位
＄0
＄0
＄0

(a)　整 型

(b) 实型
变量名 1（正 ASCII 码）
变量名 2（正 ASCII 码）
数的指数
数的尾数（高）
数的尾数↓
数的尾数↓
数的尾数（低）

(b)　实 型

变量名 1（正 ASCII 码）
变量名 2（负 ASCII 码）
字符串长度
字符串在内存的首址低八位
字符串在内存的首地高八位
$0
$0

(c)　字符串型

变量名 1（负 ASCII 码）
变量名 2（正 ASCII 码）
函数公式首址低八位
函数公式首址高八位
函数自变量表数值首址（低）
函数自变量表数值首址（高）
公式首字符

(d)　函数型

表 35-2　变量名

类　型	变 量 名	变量表内存数据
整　型	A1%	$C1，$B1
实　型	A1	$41，$31
字符串型	A1$	$41，$B1
函数型	FNA1（X）	$C1，$31

下面举例分析:
有程序:

```
10  A1 = 1024
20  A1% = 1024
30  A1$ = "ABC"
40  DEF FN A1 ( X ) = 2 * X
50  PRINT A1, A1%, A1$, FN A1 ( 1 )
```

运行此程序后，进入监控状态，检查相应的 ASCII 码:

```
*800.875
0800- 00 0D 08 0A 00 41 31 D0
0808- 31 30 32 34 00 1A 08 14
0810- 00 41 31 25 D0 31 30 32
0818- 34 00 28 08 1E 00 41 31
0820- 24 D0 22 41 42 43 22 00
0828- 38 08 28 00 B8 C2 41 31
0830- 28 58 29 D0 32 CA 58 00
0838- 4F 08 32 00 BA 41 31 2C
0840- 41 31 25 2C 41 31 24 2C
0848- C2 41 31 28 31 29 00 00
0850- 00 FF 41 31 8B 00 00 00
0858- 00 C1 B1 04 00 00 00 00
0860- 41 B1 03 23 08 00 00 C1
0868- 31 34 08 70 08 32 58 00
0870- 00 00 00 00 00 FF
```

可以看出：由 $852 至 $86D 是变量表。
$852～$858 是 A1 的变量表，其中 $41.$31 是变量名数据，$895～$85F 是

A1%的变量表，其中＄C1.＄B1是变量名数据，＄860～＄866是A1＄的变量表，其中＄41.＄B1是变量名数据，＄867～＄86D 是 FNAI (X) 变量表，其中＄C1、＄31是变量名数据。＄86E～＄874是函数自变量X的变量表。

36. FPBASIC 数组变量表的结构特点是什么

数组变量表同简单变量表一样，也是在程序运行过程中建立的，它分为整型、实型和字符串型三种。数组变量表是以一个数组为基本单位依次存放的，先定义的数组放在前面，后定义的数组放在后边。三种类型的数组表结构可参看表36-1。数组表均由表头和数组元素两部分组成，表头与数组元素部的占用内存个数为数组表总长度。表头的前两个内存单元存放变量名的ASCII码，ASCII码的特点与简单变量一样。表头变量名后边的两个内存单元存放数组表总长度，再下一个内存单元存放数组维数，以后各内存单元存放各下标定义值加1后的数值，每个下标占两个内存单元，高八位在前，低八位在后，由最后的一个下标到第一个下标依次存放。

数组元素栏中依次存放数组中各下标变量的值（即数组素值）。整型数组每个元素占两个内存单元，实型数组每个元素占5个内存单元，字符串型数组每个元素占3个内存单元，其存放特点与简单变量相似。当数组下标变量没被赋值前，各内存单元中的数值均为“0”。

表 36-1

<table>
<tr><td rowspan="10">表头</td><td rowspan="2">变量名</td><td>最高位为1 （负ASCII码）</td></tr>
<tr><td>最高位为1 （负ASCII码）</td></tr>
<tr><td rowspan="8"></td><td>数值总长低位</td></tr>
<tr><td>数值总长高位</td></tr>
<tr><td>维　　数</td></tr>
<tr><td>最后一个最大下标值加1低位</td></tr>
<tr><td>最后一个最大下标值加1高位</td></tr>
<tr><td>:</td></tr>
<tr><td>首最大下标值加1低位</td></tr>
<tr><td>首最大下标值加1高位</td></tr>
<tr><td rowspan="5">数组元素</td><td rowspan="2">0号元素</td><td>数，高位</td></tr>
<tr><td>数，低位</td></tr>
<tr><td rowspan="3">后续元素</td><td>数，高位</td></tr>
<tr><td>数，低位</td></tr>
<tr><td>:</td></tr>
</table>

(a) 整型数组变量表

表头	变量名	最高位为 0 （正 ASCII 码）
		最高位为 0 （正 ASCII 码）
		数组总长低位
		数组总长高位
		维　　数
		最后一个最大下标值加 1 高位
		最后一个最大下标值加 1 低位
		：
		首最大下标值加 1 高位
		首最大下标值加 1 低位
数组元素	0 号元素	指　　数
		尾　　数
		尾　　数
		尾　　数
		尾　　数
	后续元素	指　　数
		尾　　数
		尾　　数
		尾　　数
		尾　　数

（b）　实型数组变量表

表头	变量名	最高位为 0 （正 ASCII 码）
		最高位为 1 （负 ASCII 码）
		数组总长低位
		数组总长高位
		维　　数
		最后一个最大下标值加 1，高位
		最后一个最大下标值加 1，低位
		：
		首最大下标值加 1，高位
		首最大下标值加 1，低位
数组元素	0 号元素	字符个数
		在内存首地址高位
		在内存首地址低位
	后续元素	字符个数
		在内存首地址高位
		在内存首地址低位
		：

（c）　字符串型数组变量表

下面以一个字符串型数组为例进行分析：

程序和运行结果如下：

```
10  DIM  A1 $  ( 3, 4 )
20   FOR  I = 0  TO  3
30   FOR  J = 0  TO  4
40   READ  N $
50  A1 $  (I, J)  = N  $ ： PRINT  A1 $ ( I , J ) ; " " ;
60   NEXT  J, I
70   DATA A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P,
    Q, R, S, T
```

```
] RUN
A B C D E F G H I J K L M N O P Q R S T
```

进入监控状态后观察其数组变量表数据:

```
] CALL-151
* 6B.6E
006B- 9A 08 DF 08
* 89A. 8DF
089A- 41 B1 45 00 02 00
08A0- 05 00 04 01 5A 08 01 64
08A8- 08 01 6E 08 01 78 08 01
08B0- 5C 08 01 66 08 01 70 08
08B8- 01 7A 08 01 5E 08 01 68
08C0- 08 01 72 08 01 7C 08 01
08C8- 60 08 01 6A 08 01 74 08
08D0- 01 7E 08 01 62 08 01 6C
08D8- 08 01 76 08 01 80 08 00
```

其中，$ 89A～$ 8A2 存放表头，$ 8A3～$ 8DE 存放数组各元素的数据。各元素中字符串首字符所在内存单元的地址正是 DATA 语句中相应字符所在内存单元的地址，例如第一个元素的字符串为 A，其存放地址为 $ 85A。

```
0858- 00 83 41 2C 42 2C 43 2C
0860- 44 2C 45 2C 46 2C 47 2C
0868- 48 2C 49 2C 4A 2C 4B 2C
0870- 4C 2C 4D 2C 4E 2C 4F 2C
0878- 50 2C 51 2C 52 2C 53 2C
0880- 54 00 00 00 0A 49
```

表头中，$ 41、$ B1 为变量名，$ 45、$ 00 数组变量表总长度，$ 02 是数组维数，$ 00. $ 05 是第二个下标变量最大值加 1 的值，$ 00. $ 04 是第一个下标变量最大值加 1 的值。

37. INTBASIC 变量数据区的结构特点是什么

INTBASIC 变量数据区不同于 FPBASIC 的变量表，它有如下特点:

1. 简单变量数据、字符串变量数据和数组下标变量数据，依程序运行中前后出现的次序，顺序排列地存在内存同一区域中。这些数据依次从 $ 800 开始向高地址的内存单元存放。

2. 简单数值型变量除了变量长度不同外，其数值数据均占两个内存单元。字符串变量和数组下标变量的数据是通过 DIM 语句预留了内存空间，因而各下标变量的数据长度也是相等的。

3. 各种变量在数据区中存放的格式是一样的，都分为四栏。第一栏是变量名（或数组名），一个字符（含“$”）占一个内存单元，采用的是负 ASCII 码。第二栏存放变量名的结束符 $ 00。第三栏存放下一个变量数据存放区的首地址，它用两个内存单元，地址的低八位在前、高八位在后。第四栏是存放变量代表内容的数据。对于数值型的变量，各变量均占用两个内存单元。一个数组占用（最大下标值+1）×2 个内存单元。字符串数据占

用的内存空间为 DIM 指定的长度加 1，多余的一个内存单元用以存放一个字符串的终止符代码 $1E。各变量数据存放格式可参看表 37–1。

表 37–1　INTBASIC 程序变量数据存放格式

(a) 简单数值型变量	(b) 数组	(c) 字符串型变量	
简单数值变量名各字符的负 ASCII 码	数组名字各字符的负 ASCII 码	字符串变量名各字符的负 ASCII 码 $40("$"的代码)	名字
$00	$00	$00	名字结束符
下一个名字首字符地址低位	下一个名字首字符地址低位	下一个名字首字符地址低位	下一个名字的首地址
下一个名字首字符地址高位	下一个名字首字符地址高位	下一个名字首字符地址高位	
变量数值的低位	0 号元素的低位	各字符的负 ASCII 码，并以 $IE 为结束符	数据
变量数值的高位	0 号元素的高位		
	⋮		
	最后一个元素的低位		
	最后一个元素的高位		

举例如下:

```
>LIST
     10  DIM  A$ (13) , B$ (15) , A (4)
     20  A$ ="ABCDEFGHIJKLM"
     30  B$ ="NOPQRSTUVWXYZ"
     40  FOR  I = 1  TO  4
     50  LET  A ( I ) =I
     60  NEXT  I
     70  PRINT  A $ , B $
     80  END
```

变量数据区数据如下:

```
 * 800.83C
0800-  C1  40  00  13  08  C1  C2  C3
0808-  C4  C5  C6  C7  C8  C9  CA  CB
0810-  CC  CD  1E  C2  40  00  28  08
0818-  CE  CF  D0  D1  D2  D3  D4  D5
0820-  D6  D7  D8  D9  DA  1E  DA  C1
0828-  C1  00  36  08  00  00  01  00
0830-  02  00  03  00  04  00  C9  00
0838-  3C  08  05  00  40
```

可以看出 $ 800— $ 812 存放字符串变量 A $ 的数据。 $ C1 是变量名"A"的负 ASCII 码， $ 803、 $ 804 内存单元中存放下一个变量 B $ 数据区首地址（或说下一个变量名 B $ 首字符所在内存单元地址） $ 813。 $ C1～ $ CDR 是字符串"ABCDEFGHIJKLM"各字符相应的负 ASCII 码。 $ 813– $ 827 存放字符串 B $ 的数据。 $ 828～ $ 835 存放数组 A 的数据。 $ 836～ $ 83B 存放变量 I 的数据。

38. FPBASIC 程序的字符串数据在内存中是如何存放的

字符串型变量所代表的字符串数据有两种不同的来源。一个是源程序中由赋值语句或置数语句设定的，它存在程序区相应的内存单元中；另一个是由 INPUT 语句输入和通过字符串相加运算产生的，它存在内存空间最高地址的内存单元中(机内有 DOS 情况下，FPBASIC 程序的字符串数据由 $ 9600– $ 1 内存单元开始向低地址内存单元方向依次存放)。字符串变量表中记录下相应字符串首字符在内存单元中的地址。

下面以下述程序说明字符串数据是如何在内存中存放的。假设程序运行后，用户依次通过键盘输入 AA、BB、CC。程序如下：

```
10  A $ = "ABC"
20  FOR  I = 1  TO  3
30  INPUT  A $
40  NEXT  I
```

该程序执行第 10 语句和每次执行第 30 语句后字符串数据区首地址（ $ 6F、 $ 70)、字符串变量表、字符串数据区变化情况如表 38–1 所示。

表 38–1　字符串数据的存放过程

执行的语句		第 10 语句	第一次执行第 30 语句	第二次执行第 30 语句	第三次执行第 30 语句
键盘输入过程		–––	AA	BB	CC
指针内容	$ 6F	$ 00	$ FE	$ FC	$ FA
	$ 70	$ 96	$ 95	$ 95	$ 95
字符串变量表内容	$ 82B	$ 41	$ 41	$ 41	$ 41
	$ 82C	$ 80	$ 80	$ 80	$ 80
	$ 82D	$ 03	$ 02	$ 02	$ 02
	$ 82E	$ 09	$ FE	$ FC	$ FA
	$ 82F	$ 08	$ 95	$ 95	$ 95
	$ 830	$ 00	$ 00	$ 00	$ 00
	$ 831	$ 00	$ 00	$ 00	$ 00

续表 38-1

字符串数据区内容	$ 95FA				$ 43
	$ 95FB				$ 43
	$ 95FC			$ 42	$ 42
	$ 95FD			$ 42	$ 42
	$ 95FE		$ 41	$ 41	$ 41
	$ 95FF		$ 41	$ 41	$ 41
	$ 9600				

表中 $ 82B、 $ 82C 内存单元中存放字符串变量的变量名的 ASCII 码， $ 82D 内存单元中存放字符串变量的 A $ 的长度， $ 82E、 $ 82F 内存单元中存放字符串首字符所在内存单元的地址。

当执行了第 10 语句后， $ 82E、 $ 82F 内存单元中存放的地址为 $ 0809，它是程序中赋值语句给的字符串“ABC”首字符“A”所在内存单元的地址，这可以通过检查下面的程序区数据看出。

```
] CALL-151
* AF. B0
00AF- 2B
00B0- 08
* 800. 82B
0800-  00  0E  08  0A  00  41  24  D0
0808-  22  41  42  43  22  00  19  08
0810-  14  00  81  49  D0  31  C1  33
0818-  00  21  08  1E  00  84  41  24
0820-  00  28  08  28  00  82  49  00
0828-  00  00  05  00
```

第一次执行 INPUT A $ 语句（输入“AA”字符串），当键入 AA 并回车后，机器首先在变量表中查找有没有 A $ 变量名。如果没有，则在变量表尾建立 A $ 变量表，并填入相应数据，修改数组首地址和尾地址（各加 7），并将 0 页相应内存单元中的指针数据改变。如果变量表中有此变量名，则把新数据代换原数据。新的字符串数据存放在 $ 95FE、 $ 95FF 内存单元中，这时指针 $ 6F、 $ 70 和变量表中 $ 82E、 $ 82F 中的数据改为 $ 95FE。以后再执行 INPUT 语句，机器也做相应的工作，改变相应内存单元中的数据。

最后执行完程序，进入监控状态，检查各指针、变量表、字符串数据情况如下：

```
] RUN
? A A
? B B
```

```
?C C
] CALL-151
 * 82B, 839
082B- 41 80 02 FA 95
0830- 00 00 49 00 83 00 00 00
0838- 00 00
 * 69, 70
0069- 2B 08 39 08 39 08 FA
0070- 95
 * 95FA. 95FF
95FA- 43 43 42 42 41 41
 * 3D0G
```

39. 如何将 FPBASIC 和 INTBASIC 程序进行互换

对于较短的 BASIC 程序，可将程序列出来留在屏幕上，然后转入另一种 BASIC 语言状态（键入 INT 或 FP 命令)。进入另一种 BASIC 语言状态后，利用屏幕编辑方式用光标将屏幕中的程序走一遍，同时针对不同类型 BASIC 语言的特点将不适合新 BASIC 语言状态的语句进行适当修改。即完成了将一种 BASIC 类型的程序转换为另一种 BASIC 类型的程序。

对于较长的 BASIC，可在该种 BASIC 语言状态下将它作为程序文件存入磁盘。然后，转入另一种 BASIC 语言状态，并在此状态下用 EXEC 命令将程序文件调入内存中。再用 LIST 命令列出程序，并根据不同类型 BASIC 语言的特点将错误的语句进行适当修改。即完成了 FPBASIC 和 INTBASIC 语言的转换。

举例如下:

有 FPBASIC 程序:

```
10   FOR I = 1 TO 9
30   FOR J = 1 TO 2 * I - 1
40   PRINT I;
50   NEXT J
60   PRINT
70   NEXT I
80   FOR I = 8 TO 1 STEP - 1
100  FOR J = 1 TO 2 * I - 1
110  PRINT I;
120  NEXT J
130  PRINT
140  NEXT I
```

```
150  END
```

在上面程序的后面键入下面这段程序。

```
200  D$ = CHR $ (4)
210  PRINT D $ ;" OPEN T1 "
220  PRINT D $ ;" WRITE T1 "
230  POKE 33, 30
240  LIST 10, 150
250  PRINT D $ ;" CLOSE T1 "
```

运行补充的这段程序，建立程序文件 T1。然后进入 INTBASIC 状态，并用 EXEC 命令调出程序。

```
] RUN 200
] INT
> EXEC T1
> LIST
10   FOR I = 1 TO 9
30   FOR J = 1 TO 2 * I-1
40   PRINT I ;
50   NEXT J
60   PRINT
70   NEXT I
80   FOR I = 8 TO 1 STEP-1
100  FOR J = 1 TO 2 * I-1
110  PRINT I ;
120  NEXT J
130  PRINT
140  NEXT I
150  END

> RUN
1
2 2 2
3 3 3 3 3
4 4 4 4 4 4 4
5 5 5 5 5 5 5 5 5
6 6 6 6 6 6 6 6 6 6 6
```

```
7 7 7 7 7 7 7 7 7 7 7 7 7
8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
7 7 7 7 7 7 7 7 7 7 7 7 7
6 6 6 6 6 6 6 6 6 6 6
5 5 5 5 5 5 5 5 5
4 4 4 4 4 4 4
3 3 3 3 3
2 2 2
1
```

40. 如何使 BASIC 程序的行号重新排列

DOS 3.3 系统主盘上的 RENUMBER（重编号码）程序文件有着两个功能：改变程序行号和把两个程序连接起来。使用 RENUMBER 程序可以完成程序行号重新排列的任务。工作步骤如下：

1. 将系统主盘插入驱动器，引导 DOS 后，键入 RUN RENUMBER 命令，回车后，屏幕显示：

```
 ] RUN  RENUMBER
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&                                &
&        APPLESOFT RENUMBER&     &
&                                &
&  COPYRIGHT APPLE COMPUTER, INC 1978&  &
&                                &
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

 RENUMBER     ( DEFAULT  VALUES )

 & [ FTRST  10 ] [ , INC  10 ] [ , S  0 ] [ , E  63999 ]

 MERGE

 &H    PUT  PROGRAM  ON  HOLD
 &M    MERGE  TO  PROGRAM  ON  HOLD
 PRESS' RETURN' TO  CONTINUE…
```

表示将进入 RENUMBER 程序，并说明该程序的三个用法。最后要求按回车键。按回车键后，屏幕显示：

```
RENUMBER IS INSTALLED AND READY
IF YOU USE'FP', 'HIMEM', OR 'MAXFILES'
YOU WILL HAVE TO RE-RUN RENUMBER
```

说明 RENUMBER 文件已进入内存可以使用，如果以后操作过程中用到 FP、HIMEM 或 MAXFILES 命令，则会把 RENUMBER 程序从内存中冲掉。

2. 将要改变行号的程序引入主机

可以使用 LOAD 命令将磁盘中要改变行号的程序引入主机。

3. 键入改变行号的&命令

改变行号的&命令格式为：& [Ff] [, Ii] [, Ss] [, Ee]

该命令中的方括号内的项是可以省略的。各项的意义如下：

(1) Ff 项：f 是新程序的起始行号，无此项时 f 定为 10。F 是 FIRST 的简写。

(2) Ii 项：i 是新程序的行号间隔，无此项时 f 定为 10。I 是 INC 的简写。

(3) Ss 项：s 是原程序（待重新排行号的程序）需重排行号的某段程序的起始行号。无此项时 S 定为 0，即从原程序开始重排行号。

(4) Ee 项：e 是原程序需重排行号的某段程序的终止行号。无此项时 e 定为 63999，即至原程序末尾均重排行号。

例如，使用&命令将下面程序中第 20 至第 60 语句这段程序重排行号，新程序的起始行号为 100，行号间隔为 5。

```
10  A$="*"
20  A$="$"
30  FOR I=1 TO 10
40  PRINT TAB (15-I) ; A$
50  A $ = A $ +" $ $ "
60  NEXT I
70  END
] & F100 , I5 , S20 , E60
] LIST

10   A$=" * "
70   END
100  A$=" $ "
105  FOR I=1 TO 10
110  PRINT TAB (15-1) ; A$
```

```
115  A$ =A$+ "$ $"
120  NEXT I
```

在使用&命令时，可根据需要略去某项。举例如下:

```
10  A$ ="$"
20  FOR  I=1  TO  10
30  PRINT  TAB  (15–I)  ; A$
40  A$  =  A$  + "  $ $  "
50  NEXT  I

] &F100,   I10

] LIST

100  A$ ="$"
110  FOR  I=1  TO  10
120  PRINT  TAB  (15–I)  ; A$
130  A$  =  A$  + "$ $"
140  NEXT  I

] &F60,   I5

60  A$ ="$"
65  FOR  I=1  TO  10
70  PRINT  TAB  (15–I)  ; A$
75  A$  =  A$  + "$ $"
80  NEXT  I
```

当命令中没有I<行号间隔>项时，程序将按间隔为10自动排出。例如:

```
] &F100
] LIST

100  A$ ="$"
110  FOR  I=1  TO  10
120  PRINT  TAB  (15–I)  ; A$
130  A$  =  A$  + "  $ $  "
140  NEXT  I
```

41. 如何恢复被 NEW 命令清除的程序

当一个 BASIC 程序不小心被键入的 NEW 命令清除后，使用 LIST 命令就再也看不到程序，使用 RUN 命令也无法使程序运行了，如何将 NEW 命令清除的程序恢复呢？下面就如何恢复被 NEW 命令清除的 FPBASIC 程序和 INTBASIC 程序进行简单的分析。

1. 如何恢复被 NEW 命令清除的 FPBASIC 程序

FPBASIC 程序被 NEW 清除的实质是将程序区 $ 801 与 $ 802 内存单元的数据清为零，将 0 页 FPBASIC 指针初始化。因此要恢复被 NEW 命令清除的 FPBASIC 程序，可按下述步骤进行。

(1) 从 $ 805 内存单元开始查找第一次出现的 $ 00 字节，这就是第一个程序行的结束符。将 $ 00 字节后面的内存单元地址填入 $ 801 和 $ 802 内存单元中(低八位在前、高八位在后)。这样程序就链接上了，用 LIST 命令可以将程序列出来。但要注意，这时还不能运行、修改程序和将程序存入磁盘中。

(2) 继续往下查找程序数据，一直查到有连续三个 $ 50 出现为止，第一个 $ 00 是最末程序行的结束符，后两个 $ 00 是整个程序的结束符。三个连续 $ 00 后边的内存单元地址(或再下一个内存单元地址)即为程序的尾地址。将该地址按低八位在前，高八位在后，分别填入 $ 69、 $ 6A 内存单元中。这样，恢复的程序就可以运行和修改了。但是，这时还无法将程序存入磁盘中。

(3) 将程序尾地址按低八位在前、高八位在后，分别填入 $ AF、 $ B0 内存单元中。这样，恢复的程序就可以存入磁盘了。

举例如下：

```
10  A = 1 :  B  = 5
20  A $  =  "ABCDE" :  B  $  =  "FGHIJ"
30  FOR  I =  A  TO  B
40  A ( I )  = I :   PRINT  A ( I ) ,
50  NEXT  I
60  PRINT  A $  , B  $

] NEW
] CALL-151

* 800.860

0800-  00  00  00  0A  00  41  D0  31
0808-  3A  42  D0  35  00  27  08  14
0810-  00  41  24  D0  22  41  42  43
0818-  44  45  22  3A  42  24  D0  22
```

```
0820-  46 47 48 49 4A 22 00 32
0828-  08 1E 00 81 49 D0 41 C1
0830-  42 00 44 08 28 00 41 28
0838-  49 29 D0 49 3A BA 41 28
0840-  49 29 2C 00 4B 08 32 00
0848-  82 49 00 56 08 3C 00 BA
0850-  41 24 2C 42 24 00 00 00
0858-  0A 0A 00 00 41 80 05 15
0860-  08
*801:  0D 08

*69:  58 08
*AF:  58 08
*3D0G

] LIST
10  A = 1 :  B = 5
20  A$ = "ABCDE" : B$ = "FGHIJ"
30  FOR I = A TO B
40  A(I) = I :  PRINT A(I),
50  NEXT I
60  PRINT A$ , B $

] RUN
1                 2                 3
4                 5                 ABCDE
FGHIJ

] SAVE T-1
```

2. 用机器语言程序来恢复被 NEW 命令清除的 FPBASIC 程序

用机器语言程序来完成上述的恢复 FPBASIC 程序的工作，会使恢复工作简单、方便。下面给出两个这种机器语言程序，它们均存入以 $ 0300 开始的内存单元中。使用时，可将它从磁盘调入主机内存中，然后键入：

] CALL 768

即可迅速完成恢复被 NEW 命令清除的 FPBASIC 程序。

第一个机器语言程序清单如下:

```
0300-  A2 05      LDX  #$05
0302-  BD 00 08   LDA  $0800,X
0305-  F0 03      BEQ  $030A
0307-  E8         INX
0308-  D0 F8      BNE  $0302
030A-  18         CLC
030B-  8A         TXA
030C-  69 01      ADC  #$01
030E-  8D 01 08   STA  $0801
0311-  8D 1F 03   STA  $031F
0314-  A9 08      LDA  #$08
0316-  8D 02 08   STA  $002
0319-  8D 20 03   STA  $0320
031C-  A2 00      LDX  #$00
031E-  BD 23 08   LDA  $0823,X
0321-  95 2A      STA  $2A , X
0323-  D0 07      BNE  $032C
0325-  E0 01      CPX  #$01
0327-  F0 14      BEQ  $033D
0329-  E8         INX
032A-  D0 F2      BNE  $031E
032C-  E0 01      CPX  $#$01
032E-  D0 F9      BNE  $0329
0330-  A5 2A      LDA  $2A
0332-  8D 1F 03   STA  $031F
0335-  A5 2B      LDA  $2B
0337-  8D 20 03   STA  $0320
033A-  4C 1C 03   IMP  $031C
033D-  18         CLC
033E-  AD 1F 03   LDA  $031F
0341-  69 02      ADC  #$02
0343-  85 69      STA  $69
0345-  85 6B      STA  $6B
0347-  85 6D      STA  $6D
0349-  85 AF      STA  $AF
034B-  AD 20 03   LDA  $0320
034E-  69 00      ADC  #$00
0350-  85 6A      STA  $6A
0352-  85 6C      STA  $6C
0354-  85 6E      STA  $6E
0356-  85 B0      STA  $B0
0358-  60         RTS
```

第二个机器语言程序清单如下:

```
0300-  AD 02 08   LDA  $0802
0303-  D0 50      BNE  $0355
0305-  A9 08      LDA  #$08
0307-  85 6A      STA  $6A
0309-  A9 01      LDA  #$01
030B-  85 69      STA  $69
030D-  A0 04      LDY  #$04
030F-  C8         INY
0310-  20 51 03   JSR  $0351
0313-  B1 69      LDA  ($69),Y
0315-  D0 F8      BNE  $030F
0317-  C8         NY
0318-  20 51 03   JSR  $0351
031B-  C8         INY
031C-  20 51 03   JSR  $0351
031F-  8C 01 08   STY  $0801
0322-  A4 6A      LDY  $6A
0324-  8C 02 08   STY  $0802
0327-  A0 05      LDY  #$05
0329-  84 69 69   STY  $69
032B-  A0 08      LDY  #$08
032D-  84 6A      STY  $6A
032F-  A0 00      LDY  #$00
0331-  A2 03      LDX  #$03
0333-  E6 69      INC  $69
0335-  20 51 03   JSR  $0351
0338-  B1 69      LDA  ($69),Y
033A-  D0 F5      BNE  $0331
033C-  CA         DEX
033D-  D0 F4      BNE  $0333
```

```
033F-  E6 69     INC  69
0341-  20 51 03  JSR  $0351
0344-  E6 69     INC  $69
0346-  20 51 03  JSR  $0351
0349-  A5 69     LDA  $69
034B-  85 AF     STA  $AF
034D-  A5 6A     LDA  $6A
034F-  85 B0     STA  $B0
0351-  D0 02     BNE  $0355
0353-  E6 6A     INC  $6A
0355-  60        RTS
```

使用机器语言恢复被 NEW 命令清除的 FPBASIC 程序实例如下:

```
10 A $ = "*"
20 FOR I = 1 TO 10
30 PRINT TAB (15 - I);; A$
40 A$ = A$ + "*"
50 NEXT I

] NEW

] LIST

] BLOAD NEW-1
] CALL768
] LIST
10 A$ = "*"
20 FOR I = 1 TO 10
30 PRINT TAB (15-I);; A$
40 A$ = A $ + "*"
50 NEXT I
```

```
] RUN
                  *
                 **
                ***
               ****
              *****
             ******
            *******
           ********
          *********
         **********
```

注: NEW-1 是磁盘中的机器语言程序名。

3. 如何恢复被 NEW 命令清除的 INTBASIC 程序

INTBASIC 程序是存在高地址的内存单元中，其程序尾地址总指向 $9600(0 页 $4C、$CD 内存单元存放 INTBASIC 程序尾地址)，程序起始地址随程序长度的变化而改变，$CA、$CB 内存单元中存放 INTBASIC 程序的起始地址。一旦键入 NEW 命令后，机器将程序首地址指针($CA、$CB)和数据区尾地址指针($CC、$CD)恢复到初始化状态，即 $CA、$CB 内存单元存放 $9600，$CC、$CD 内存单元存放 $0800，而 INTBASIC 程序数据没有任何变化。因此，要恢复被 NEW 命令清除的 INTBASIC 程序，只需将程序区首地址按低八位在前、高八位在后填入 $CA、$CB 内存单元中即

可。

INTBASIC 程序起始地址的查找可根据 INTBASIC 程序数据存放特点，由 $9600 内存单元开始向低单元开始向低地址方向查找。关于 INTBASIC 程序数据存放的特点可参看"INTBASIC 程序在内存程序区中是如何存放的"问题的解答。

举例如下:

```
5  DIM A$(5)
10 A = 1
20 B = 2
30 A$ = "ABCDE"
40 PRINT A,B,A$
50 END

>NEW

>CALL-151

* 95B5.9600

95B5- 00 00 00
95B8- 00 00 00 00 00 00 00 00
95C0- 00 00 00 00 0C 05 00 4E
95C8- C1 40 22 B5 05 00 72 01
95D0- 09 0A 00 C1 01 B1 01 00
95D8- 01 09 14 00 C2 71 B2 02
95E0- 00 01 0E 1E 00 C1 40 70
95E8- 28 C1 C2 C3 C4 C5 29 01
95F0- 0B 28 00 62 C1 49 C2 48
95F8- C1 40 01 05 32 00 51 01
9600- 62
* CA : C4 95

* 3D0G

> LIST
5  DIM A$ (5)
10 A = 1
20 B = 2
30 A$ = "ABCDE"
40 PRINT A,B,A$
50 END

>RUN
1       2       ABCDE
```

>SAVE T-1

42. 如何使计算机启动 DOS 时不破坏内存中的 BASIC 程序

当你花费不少时间通过键盘输入一个较长的 BASIC 程序，正准备将它存入磁盘时，突然发现没有启动 DOS。这时，因机内没有 DOS，所以你无法将程序存入磁盘，而要启动 DOS 又会破坏机内的 BASIC 程序。如何使计算机启动 DOS 时不破坏内存中的 BASIC 程序呢？要解决这个问题首先应了解为什么键入 PR#6 命令启动 DOS 后会破坏机内的 BASIC 程序。

键入 PR#6 后，计算机会按下述步骤来引导 DOS。

1. 执行驱动器卡上的机器语言程序，将磁盘 0 磁道 0 扇区中的引导程序 1 读入机内 $ 800～ $ 8FF 内存单元中。

2. 执行引导程序 1，将磁盘 0 磁道 1～9 扇区的引导程序 2 读入机内，同时也再将引导程序 1 读入机内。它们存入 $ 3600～ $ 4000 内存单元中。

3. 执行引导程序 2，将磁盘中 0 磁道 A～B 扇区的再分配程序读入机内 $ 1B00～ $ 1D00 内存单元中，并将 0 磁道 C 扇区到 2 磁道 4 扇区的 DOS 读入 $ 1D00～ $ 4000 内存单元中。

4. 执行再分配程序，将 $ 1D00～ $ 4000 内存单元中的 DOS 移至 $ 9D00～ $ C000 内存单元中。同时将 $ 9600～ $ 9D00 内存区域置为文件缓冲区。

5. 将主机第 0 页和第 3 页有关 BASIC 指针和 DOS 指针等初始化。同时将磁盘中的 BASIC 语言欢迎程序读入主机以 $ 800 为开始的内存单元中，并执行该程序。

上述的工作步骤可参看表 42-1。由表 42-1 可以看出来，启动 DOS 后会破坏机内的 BASIC 程序和 0 页的有关指针数据。要使机器在启动 DOS 时不破坏 BASIC 程序，需将 BASIC 程序和 0 页的指针数据移至不会因起动 DOS 而受破坏的内存"安全"区域中。"安全"的内存区域可选定为 $ 4000～ $ 9600。只要程序长度不超过这个范围，即可得到保护(即使超过了，也可以保护相当长的一部分 BASIC 程序)。

表 42-1 DOS 启动过程

地址	内容	
$ C000～$ 9D00	DOS	最终的 DOS
$ 9D00～$ 9600	文件缓冲区	
$ 9600～$ 4000		
$ 4000～$ 3600	引导程序 2	步骤 2
$ 3600～$ 1D00	DOS	步骤 2
$ 1D00～$ 1B00	再分配程序	
$ 1B00～$ 900		
$ 900～$ 800	引导程序 1	步骤 1

一. 如何使计算机启动 DOS 时不破坏内存中的 FPBASIC 程序

对于 FPBASIC 程序，根据它的存放特点和指针特点，其工作步骤如下:

(1) 进入监控状态并读出 FPBASIC 程序的尾地址(即 $ AF、 $ B0 中的内容)，计算程序长度:

$ 程序长度 = $ 程序尾地址 − $ 800 + $ 1

(2) 将 BASIC 程序移至“安全”区域:

```
*  4000  <  800 · 程序尾地址 M
```

(3) 保护 0 页 BASIC 指针数据:

```
*  9060  <  60 · FFM
```

(4) 启动 DOS:

```
* 6  CTRL-P
```

(5) 移回 BASIC 程序

```
*  800  <  4000 · 4000+程序长度 M
```

(6) 移回 0 页 BASIC 指针数据:

```
*  60  <  9060 · 90FFM
```

(7) 回到 FPBASIC 工作状态，列出程序清单，并将程序存入磁盘。

```
*  3D0G
] LIST
] SAVE  <文件名>
```

举例如下:

```
10  A$  =  " * "
20  FOR  I  =  1  TO  10
30  PRINT  TAB ( 15-I ) ; ; A$
40  A$  =  A$  +  " * "
50  NEXT  I

] CALL-151

*  AF.B0

00AF-  40
00B0- 08
*4000  <  800 . 840  M
*9560  <  60 . FF  M

*6  CTRL-P
```

引导 DOS 后，继续执行:

```
*800  <  4000 . 4040  M
*60  <  9560 . 95FFM
*3D0G
] LIST
10  A$  =  " * "
20  FOR  I =  1  TO  10
30  PRINT  TAB ( 15-I ) ; A$
40  A$  =  A$  +  " * "
50  NEXT  I

] SAVE T-1
```

二. 如何使计算机启动 DOS 时不破坏内存中的 INTBASIC 程序

由于 INTBASIC 程序存放在高地址的内存单元中(从 $95FF 开始向低地址内存单元方向存放)，所以在引导 DOS 后只需将程序首地址指针($CA、$CB)数据复原即可。这与恢复被 NEW 命令清除方法一样。

举例如下:

```
>LIST
10  DIM  A$ ( 10 )
20  A$  =  " ABC6DEFGH6IJ "
30  A  = 1 : B  = 1
40  PRINT  A , B , A$
50  END
>CALL-151
*CA . AB
00CA-  C2  95
*6CTRL-P
] INT
>  LIST
>CALL-151
*CA : C2  95
*3D0G
>LIST
10  DIM  A$ ( 10 )
20  A$  =  " ABCDEFGHIJ "
30  A  =  1 : B  =  1
40  PRINT  A , B , A$
50  END
>RUN
1          1            ABCDEFGHIJ
>SAVE  T - 1
```

43. 如何修改 CTRL-RESET 的控制功能

中华学习机进入 FPBASIC 工作状态，有两种启动方式，一种是冷启动，一种是热启动。所谓冷启动是指中华学习机接通电源到进入 FPBASIC 状态这一过程。所谓热启动是指按下 CTRL-RESET 键到进入 FPBASIC 状态这一过程。

在冷启动过程中，中华学习机完成一系列工作：置字符正常显示方式；置屏幕为文本显示文式；输出设备定为显示器；输入开关定为键盘；使扬声器发出一下声响；建立已接通电源的标志；清内存；设置监控常量；检查八个槽口；判断是否有磁盘驱动器(若没有则引导 DOS)；将内存初始化并建立内存初始化标志。然后进入 FPBASIC 工作状态。在完成内存初始化工作时，将 $03F2～$03F4 三个内存单元置数。前两个内存单元按照低位在前高位在后的次序，存入按下 CTRL-RESET 键后的转向地址。在无 DOS 时，$03F2 内存单元中存入 $0B，$03F3 内存单元中存入 $E0，$E003 是 FPBASIC 的入口地址。在启动 DOS 时，$03F2、$03F3 内存单元中分别存入 $BF、$9D，$9DFB 是 DOS 的入口地址。而 $03F4 内存单元中存放的是 $03F3 内存单元中的数与 $A5 异或后的值，未启动 DOS 时为 $45，启动 DOS 时为 $38。

所谓异或运算是指两数转换为二进制数后，相同位的数进行异或逻辑运算。异或逻辑运算的结果是 (+为异或逻辑运算符):

$1+1=0$　　　$0+0=0$　　　$1+0=1$　　　$0+1=1$

例如:　　　$ E0＝1110000、 $ A5＝10100101

```
      11100000
   +  10100101
  ──────────────
      01000101     → $ 45
```

热启动过程，中华学习机只终止正在进行的工作，完成置字符正常显示方式，输出设备定为显示器，输入设备定为键盘，使扬声器响一声等待工作，并不清内存和进行内存初始化。

当按下 CTRL−RESET 键后，计算机将 $ 03F3 与 $ 03F4 内存单元中的内容进行异或运算，看其值是否为 $ A5。如果是 $ A5，则转至 $ 03F2 与 $ 03F3 内存单元指示的地址去执行程序；如果不是 $ A5，则执行热启动。通常，因 $ 03F3 与 $ 03F4 内存单元中的数异或后均为 $ A5，则会转至 $ 03F2 与 $ 03F3 指示的地址 $ E003(或 $ 9DBF)，即转至 FPBASIC 工作状态。所以，通常 CTRL−RESET 键的功能是完成一个热启动过程。

依据上述工作原理，适当改变 $ 03F2～ $ 03F4 内存单元中的数据，即可改变 CTRL−RESET 键的功能，举例如下:

1. 把热启动改为冷启动

如果将 $ 03F2～ $ 03F4 内存单元中一个数据改变了，使 $ 03F3 与 $ 03F4 内存单元中的数异或后不等于 $ A5，即可使按下 CTRL−RESET 键后，计算机不是执行热启动，而是执行冷启动，好象刚刚接通电源开关一样。

2. 转入非冷启动的其它工作状态

如果将 $ 03F2～ $ 03F4 内存中的数据改变，但仍保持 $ 03F3 与 $ 03F4 内存单元中的数异或后为 A $，则按 CTRL−RESET 键后，计算机会转至新的地址(非 $ E003 或 $ 9DBF)去工作。例如:

(1). 转至监控状态

＊03F2 ：69　FF　5A

＊FF69 是监控的入口地址。

(2). 转至锁定状态

＊03F2 ：F2　03　A6

按 CTRL−RESET 键后，计算机“嘟”的一声，即进入锁定状态，不再做任何工作。

(3). 重新运行 BASIC 程序

＊03F2 ：66　D5　70

(4). 清除 BASIC 程序

＊03F2 ：4B　D6　73

(5). 清屏幕为黑色

＊03F2 ：58　FC　59

在完成上述工作时，特别要注意计算 $ 03F4 内存单元中的数应使其与 $ 03F3 内存单元中的数异或后为 $ A5。这一运算较麻烦，可交计算机来完成。在给 $ 03F2 和 $ 03F3 内存置好数后，只需执行 CALL−1169，计算机即可自动给 $ 03F4 置好数。

由上述分析可看出，改变 CTRL-RESET 键功能后，可达到将 BASIC 程序保密的目的。

44. 如何使 BASIC 程序无法修改

FPBASIC 程序规定程序行行号的最大值是 63999，键盘输入的行号大于 63999 时，机器不接受。如果要使程序的行号大于 63999，可在监控状态下，将程序区放行号的内存单元中的数据做相应的改动即可。例如要将第一个程序行的行号改为 65533，可将 $ 803 和 $ 804 内存单元中的数改为 $ FD 和 $ FF，$ FFFD 的十进制表示即为 65533。行号大于 63999 的程序行的程序，用户无法进行修改，而运行不受影响。

举例如下：

```
10  A$  =  1 : B  =  2
20  A$  =  " $ "
30  PRINT  A , B , A$
] CALL-151

* AF . B0

00AF-  27
00B0-  08
* 800 . 827

0800-  00  0D  08  0A  00  41  D0  31
0808-  3A  42  D0  32  00  18  08  14
0810-  00  41  24  D0  22  24  22  00
0818-  24  08  1E  00  BA  41  2C  42
0820-  2C  41  24  00  00  00  0A  22
```

```
* 81A : FF  FF

* 80F : FE  FF

* 803 : FD  FF

* 3D0G

] LIST

65533  A  =  1 : B  =  2
65534  A$  =  " $ "
65535  PRINT  A , B , A$

] RUN
1              2              $
```

45. 如何给 BASIC 程序进行简单的保密

一个较好的程序，往往要花费程序设计者大量的精力。为了不让别人查看自己的程序，可对程序进行保密。下面介绍一些简单的对 BASIC 程序进行保密的方法。

1. 修改源程序链指针

(1) * 801:00 或 POKE 2049，0

修改后不管你的程序有多少，LIST 命令键入后屏幕上只有第一行程序显示出来，其它程序行均不显示，但运行程序和存贮程序不受影响。其原因是，经此修改后 $ 800、$ 801 内存单元中均为 $ 00，使机器列程序时误认为程序结束，而运行程序是以连续三个 $ 00 为结束点。

(2) *801：01 或 POKE 2049，1

修改后，LIST 命令使屏幕上重复显示第一行程序。其原因是第一个链指针总指向第一个程序行。

(3) 如果使屏幕上连第一个程序行也不显示，可采用下述方法：

首先在你的程序行前加一行程序：

0 PRINT "ABCDEFGHIJ

应注意十个字符，之后加十个空格，空格后无引号。

然后，对内存单元进行相应的修改：

*807：08 08 08 08 08 08 08 08 08 08 08 08

*801：00

回到 BASIC 状态后，再用 LIST 命令列程序，屏幕上没有任何程序行显示出来，而程序运行仍正常。

举例如下：

```
0   PRINT "ABCDEFGHIJ
10  A = 1：B = 2
20  A$ = " $ "
30  FOR I = 1 TO 5
40  PRINT A，B，：PRINT A$
50  A = B + I：B = A + I
60  A$ = A$ + A$
70  NEXT I
] CALL-151
*AF.BO
00AF- 72
00B0- 08
*800.872

0800- 00 1C 08 00 00 BA 22 41
0808- 42 43 44 45 46 47 48 49
0810- 4A 20 20 20 20 20 20 20
0818- 20 20 20 00 28 08 0A 00
0820- 41 D0 31 3A 42 D0 32 00
0828- 33 08 14 00 41 24 D0 22
0830- 24 22 00 3E 08 1E 00 81
0838- 49 D0 31 C1 35 00 4B 08
0840- 28 00 BA 41 2C 42 3A BA
0848- 41 24 00 5B 08 32 00 41
0850- D0 42 C8 49 3A 42 D0 41
0858- C8 49 00 68 08 3C 00 41
0860- 24 D0 41 24 C8 41 24 00
0868- 6F 08 46 00 82 49 00 00
0870- 00 0A 53
*807：08 08 08 08 08 08 08 08 08 08
*801：00
*3D0G
```

上述保密的解密方法是恢复链指针。

2. 禁止输入命令

*D6：FF 或 POKE 214， 225

*D6：80 或 POKE 214， 128

$D6 内存单元是存放 BASIC 程序保护标志的，通常的值为 $00。改变 $D6 内存单元中的数值后，不管你按什么键，输入什么命令，机器均视为 RUN 命令而使程序重新执

行，使外人无法窥视程序内容，起到保密作用。

举例如下：

```
10 A = 1 : B = 2
20 A$ = " $ "
30 FOR I = 1 TO 5
40 PRINT A , B : PRINT A$
50 A = B + I : B = A + I
60 A$ = A$ + A$
70 NEXT I
```

```
] POKE 214 , 128

] LIST
1              2
$
3              4
$ $
6              8
$ $ $ $
11             14
$ $ $ $ $ $ $ $
18             22
$ $ $ $ $ $ $ $ $ $ $ $ $ $ $ $
```

```
] POKE 214 , 0
1              2
$
3              4
$ $
6              8
$ $ $ $
11             14
$ $ $ $ $ $ $ $
18             22
$ $ $ $ $ $ $ $ $ $ $ $ $ $ $ $
```

3. 改变输出信息显示方式

＊32 : 80　或　POKE 50, 128

修改后，会造成所有输出信息(键盘输出除外)均为空格。$32内存单元存放显示方式代码，正常显示时其值为$FF，改变它的值，会产生各种奇怪的现象，使他人无法用LIST命令观察程序内容。要恢复正常可键入：

＊32 : FF　或　POKE 50, 255

4. 重新启动DOS

在你的程序的末尾加入一个语句：

PRINT CHR $ (4) "PR#6"

可在程序运行结束后重新启动DOS，破坏源程序，以达到保密的目的。

5. 防止程序中断

一个正在运行的BASIC程序，如果发生语法错误或操作错误，以及键CTRL−RESET或CTRL−C键，会使程序停止运行，其后可以用LIST命令观看程序内容。要对程序保密必须针对上述情况防止中华学习机发生中断。具体方法如下：

(1) 防止因语法或操作错误以及按 CTRL—C 键产生中断

在 FPBASIC 语言中，按 CTRL—C 键被认为是一种错误形式，错误代码为 $FF (255)。因此，防止程序被 CTRL—C 实质与防止因语法或操作错误产生中断是一回事。为此，我们可在要保密的程序中加入以下语句:

```
0 ONERR GOTO 63999
63999 GOTO 63999
```

当产生各种错误（含按 CTRL—C 键）时，程序转至第 63999 程序行运行，造成死循环，防止程序中断。

(2) 防止程序被 CTRL—RESET 中断

在要保密的程序中加入以下语句:

```
1 POKE 1010, 48 : POKE 1011, 3 : CALL-1169
2 POKE 768, 76 : POKE 769, 0 : POKE770, 3
```

第 1 程序行的作用是将 CTRL—RESET 控制转至 $ 300 处，去执行一个死循环程序(JMP $ 300)。$ 300～ $ 302 内存单元存放 JMP $ 300 程序。当按下 CTRL—RESET 键后，机器处于死循环状态。关于 CTRL—RESET 的控制作用可参看问题“如何修改 CTRL—RESET 的控制功能”的解答。

为了不使他人窥视你的程序，可将上述保密方法与其它程序加密方法综合在一起使用，可达到更为理想的效果。

46. 如何使 LIST 命令后程序中没有空格

在我们编写和修改程序中，往往希望程序中没有多余的空格(不含字符串中的空格)。尤其是在修改 DATA、REM 语句和引号内的字符串时，每行右边的空白会因光标扫过而使程序增加更多的无用的空格。运行下面的机器语言程序(* 300G 或 CALL-768)，可使你用 LIST 命令列程序时，程序中没有多余的空格。

```
0300-  18        CLC                0314-  D0 04      BNE   $031A
0301-  A9 D0     LDA   #$D0         0316-  E6 07      INC   $07
0303-  A0 00     LDY   #$00         0318-  F0 05      BEQ   $031F
0305-  85 07     STA   $07          031A-  2C 82 C0   BIT   $C082
0307-  84 06     STY   $06          031D-  90 EA      BCC   $0309
0309-  B1 06     LDA   ($06),Y      031F-  A9 01      LDA   #$01
030B-  2C 83     C0]BIT $C083       0321-  8D FD D6   STA   $D6FD
030E-  2C 83     C0]BIT $C083       0324-  8D 51 D7   STA   $D751
0311-  91 06     STA   ($06),Y      0327-  8D 63 D7   STA   $D763
0313-  C8        INY                032A-  A9 40      LDA   #$40
```

要回到正常状态，可在监控状态下键入:

```
* C082
```

或在 BASIC 状态键入；

    POKE 800, 32 : CALL768

如果要从正常状态再进入无空格状态，可键入：

    POKE 800, 1 : CALL768

举例如下：

```
10  PRINT ″HHJLKJLE    UKKU″
20  PRINT     ″KLJKLKL″

] LIST

10 PRINT ″HHJLKJLK    UKKU″
20 PRINT     ″KLJKLKL″

] POKE 800 , 32 : CALL768
] LIST

10 PRINT ″HHJLKJLK    UKKU″
20 PRINT     ″KLJLKL″
```

如果键入 POKE 800, n : CALL768（n 取 0～127 之间的整数），则用 LIST 命令列程序时，以 CHR $(n)取代原程序各词间的空格。例如：

```
] LIST
10 PRINT   ″GKHKHJKLK″
20 PRINT ″HLJKLJ   UIOLUPO″

] POKE 800 , 32 : CALL768

] LIST

10   PRINT     ″GKHKHJKLK″
20   PRINT ″HLJKLJ    UIOLUPO″

] POKE 800 , 65 : CALL768

] LIST

10 AAPRINTA   ″GKHKHJKLK″
20 AAPRINTA ″HLJKLJ    UIOLUPO                ″

] CALL-151

* C082

C082-  00
] 3D0G

] LIST

0    PRINT   ″GKHKHJKLK″
20   PRINT   ″HLJKLJ    UIOLUPO″
```

对于没有 16KRAM 扩展卡的 APPLE-Ⅱ微机，则可利用下面程序。

```
  * 300 . 322
0300-   C9  A2       CMP   #$A2
0302-   D0  37       BNE   $031B
0304-   A9  0D       LDA   #$0D
0306-   85  36       STA   $36
0308-   A9  A2       LDA   #$A2
030A-   4C  F0  FD   JMP   $FDF0
```

```
030D-  C9 A2      CMP  #$A2       0318-  4C 08 03   JMP  $0308
030F-  F0 03      BEQ  $0314      031B-  C9 A0      CMP  #$A0
0311-  4C F0 FD   JMP  $FDF0      031D-  D0 01      BNE  $0320
0314-  A9 00      LDA  #$00       031F-  60         RST
0316-  85 36      STA  $36        0320-  4C F0 FD   JMP  $FDF0
```

使用时，键入：

POKE 33，33：POKE54，0：POKE 55，3：CALL1002

退出此状态，还原回正常状态，只需按 CTRL-RESET 键。

47. 如何使中华学习机能够用英文小写字母编程序

通常中华学习机只能接受用英文大写字母编写的程序和命令。运行下面的程序后，可以使中华学习机接受英文小写字母编写的程序和命令。键入 POKE6，0 后为英文大写字母显示，键入 POKE6，1 后为英文小字母。键入 CTRL-RESET 键后倒使机器失去此功能，要恢复其功能，必需重新运行下面的程序。运行机器语言程序的方法是：

在 BASIC 语言状态　　键入 CALL768

在监控状态：　　键入 * 300G

程序如下：

```
*300.326
0300-  A9 0B      LDA  #$0B       0314-  C9 DB      CMP  #$DB
0302-  85 36      STA  $36        0316-  10 03      BPL  $031B
0304-  A9 03      LDA  #$03       0318-  18         CLC
0306-  85 37      STA  $37        0319-  65 08      ADC  $08
0308-  6C F2 03   JMP  ($03F2)    031B-  4C F0 FD   JMP  $FDF0
030B-  48         PHA             031E-  A5 06      LDA  $06
030C-  20 1E 03   JSR  $031E      0320-  F0 02      BEQ  $0324
030F-  68         PLA             0322-  A9 20      LDA  $06
0310-  C9 C1      CMP  #$C1       0324-  85 08      STA  $08
0312-  30 07      BMI  $031B      0326-  60         RST
```

48. 如何将光标改为一条闪烁的横线

运行下面这个机器语言程序，即可使屏幕的光标由闪烁的小方块改为闪烁的横线，可以减轻人眼的疲劳感。如果主机内没有 DOS，需将机器语言程序中 $316 内存单元的值改为 $03，将 $317 内存单元的值改为 $E0。

```
*300L
0300-  A9 0D        LDA  #$0D
0302-  8D F2 03     STA  $03F2
0305-  A9 03        LDA  #$03
0307-  8D F3 03     STA  $03F3
030A-  20 6F FB     SR   $FB6F
030D-  A9 18        LDA  #$18
030F-  85 38        STA  $38
0311-  A9 03        LDA  #$03
0313-  85 39        STA  $39
0315-  4C D0 03     JMP  $03D0
0318-  48.          PHA
0319-  86 06        STX  $06
031B-  A9 DF        LDA  $DF
031D-  91 28        STA  ($28),Y
031F-  20 2F 03     ISR  $032F
0322-  30 1D        BMI  $0341
0324-  68           PLA
0325-  91 28        STA  ($28),Y
0327-  48           PHA
0328-  20 2F 03 JSR $032F
032B-  30 14        BMI  $0341
032D-  10 EC        BPL  #031B
032F-  A2 50        LDX  #$50
0331-  A9 FF        LDA  #$FF
0333-  2C 00 C0BIT  $C000
0336-  30 08        BMI  $0340
0338-  38           SEC
0339-  E9 01        SBC  #$01
033B-  D0 F6        BNE  $0333
033D-  CA           DEX
033E-  D0 F1        BNE  $0331
0340-  60           RTS
0341-  68           PLA
0342-  A6 06        LDX  $06
0344-  4C 26 FDJMP  $FD26
```

49. 如何有趣地清文本屏幕

在BASIC程序中，经常需要清除不用的文本屏幕，一般可用HOME语句来完成。但是，为了增加程序的趣味性与艺术性，可以采用特殊的有趣的清屏幕方法。下面介绍几种有趣地清文本屏幕的方法。

1. 从下至上一行行清屏幕。

```
5  REM PROGRAM 49.1
10 FOR I = 1024 TO 2039
20 POKE I , 170
30 NEXT I
40 GET A$
50 FOR Y = 24 TO 1 STEP - 1
60 VTAB Y : CALL 64668
70 FOR I = 1 TO 100 : NEXT J
80 NEXT Y
```

2. 自上而下一行行清屏幕

```
5  REM PROGRAM 49.2
10 FOR I = 1024 TO 2039
20 POKE I , 170
30 NEXT I
40 GET A$
50 FOR Y = 1 TO 24
60 VTAB Y : CALL 64668
70 FOR J = 1 TO 100 : NEXT J
80 NEXT Y
```

3. 从中间向左右两边清屏幕。

```
5   REM PROGRAM 49.3
10  FOR I = 1024 TO 2039
20  POKE I , 170
30  NEXT I
40  GET A $
50  FOR I = 1 TO 20
60  POKE 32,20 - I : POKE 33,2 *
    I :HOM
70  FOR J = 1 TO 100 : NEXT J
80  NEXT I
```

4. 从中间向上下两边清屏幕。

```
5   REM PROGRAM 49.4
10  FOR I = 1024 TO 2039
20  POKE I , 170
30  NEXT I
40  GET A $
50  FOR I = 1 TO 12
60  POKE 34 , 12 - I : POKE
    35 , 12 + I : HOME
70  FOR J = 1 TO 100 : NEXT J
80  NEXT I
```

5. 将屏幕分为四部分，每部分六行字符，每次清各部分的一行，自上至下象拉起百叶窗一样地清屏幕。

```
5    REM PROGRAM 49.5
10   FOR I = 1024 TO 2039
20   POKE I , 170
30   NEXT I
40   GET A $
50   L = - 5
60   FOR I = 0 TO 5
70   K = 1 TO 19 STEP 6
80   VTAB I + K : CALL 64668
90   FOR J = 1 TO 30 : NEXT J
100  NEXT K , I
```

6. 从中间向四周清屏幕。

```
5   REM PROGRAM 49.6
10  FOR I = 1024 TO 2039
20  POKE I , 170
30  NEXT I
40  GET A $
50  FOR I = 1 TO 12
60  POKE 32 , 12 - I : POKE 33 , I * 2 + 16 : POKE 34 , 12 - I :
POKE 35 , 12 + I : HOME
```

```
70  FOR J = 1 TO 50 : NEXT J
80  NEXT I
```

7. 从上至下水平光柱清屏幕。

```
5    REM  PROGRAM 49.7
10   FOR I = 1024 TO 2039
20   POKE I , 170
30   NEXT I
40   GET A$
50   FOR Y = 1 TO 24
60   VTAB Y : INVERSE : PRINT SPC ( 40 )
70   FOR J = 1 TO 30 : NEXT J
80   VTAB Y : NORMAL : PRINT SPC ( 40 )
90   NEXT Y
100  HOME
```

8. 从下至上用水平行光柱清屏幕。

```
5    REM  PROGRAM 49.8
10   FOR I = 1024 TO 2039
20   POKR I , 170
30   NEXT I
40   GET A$
50   FOR Y = 24 TO 1 STEP - 1
60   VTAB Y : INVERSE : PRINT SPC ( 40 )
70   FOR J = 1 TO 30 : NEXT J
80   VTAB Y : NORMAL : PRINT SPC ( 40 )
90   NEXT Y
100  HOME
```

9. 使屏幕上的字符象雪花一样从上至下地坠落直至屏幕清为黑色。将下面的机器语言程序存入磁盘（设名子为 S-9），

```
0300- A5 22 48 A9 00 A8 85 06
0308- A9 D0 85 07 A5 4E 18 69
0310- 0B 71 06 C8 85 4E 38 E5
0318- 21 B0 F9 A5 4F 18 69 0D
0320- 71 06 18 65 22 85 4F 38
0328- E5 23 B0 F6 A4 4E 84 24
0330- A5 4F C5 22 90 EC 20 5B
0338- FB B1 28 C9 A0 F0 16 48
0340- A9 A0 91 28 A5 25 C9 17
0348- 90 03 68 B0 08 E6 25 20
```

```
0350-  22  FC  68  91  28  A0  00  84        0368-  F5  E6  22  A5  22  C5  23  90
0358-  24  A5  22  20  5B  FB  B1  28        0370-  E4  68  85  22  60  E6  06  D0
0360-  C9  A0  D0  11  C8  C4  21  90        0378-  93  E6  07  D0  8F  F0  89  00
```

然后，用 DOS 命令调用它。例如:

```
5   REM  PROGRAM  49.9
10  FOR  I  =  1024  TO  2039
20  POKE  I , 170
30  NEXT  I
40  GET  A$
50  PRINT  CHR$ ( 13 ); CHR$ ( 4 ) " BRUN  S-9 , A$300 "
```

50. 如何将机器语言程序转换成 BASIC 程序

使用下面的程序可将存放在内存区域中的机器语言程序转换成 EXEC 程序文件。然后使用 EXEC 命令即可得到相应的 BASIC 程序。

运行该程序后，需输入存放机器语言的内存区域首地址 (START ADDRESS) 和尾地址 (END ADDRESS)，形成的 BASIC 语言的第一个行号 (FIRST LEBEL)和行号间隔 (LEBEL INTERVAL)，还有形成的顺序文件 (即 EXEC 程序文件) 文件名 (FILE NAME)。然后机器自动建立含有内存区域所有数据的顺序文件。

举例如下:

机器语言在内存区域中首址为 768，尾址为 900，相应的 BASIC 程序第一个行号为 100，行号间隔为 5，顺序文件名为 ABC。

```
5   REM  PROGRAM  50.1
10  HOME : VTAB  8 : HTAB  10
15  INPUT " START  ADDRESS : " ; A : PRINT
20  HTAB  10 : INPUT " END  ADDRESS : " ; B : PRINT
25  HTAB  10 : INPUT " FIRST  LEBEL : " ; C : PRINT
30  HTAB  10 : INPUT " LEBEL  INTERVAL : " ; D : PRINT
35  HTAB  10 : INPUT " FILE  NAME : " ; F$
40  C0 = C : C1 = C + D : C2 = C1 + D : C = C2 + D
50  D$ = CHR$ ( 13 ) + CHR$ ( 4 )
60  PRINT  D$ ;" OPEN " : 6F$ : PRINT  D  $ ;" DELDTD " ; F$
70  PRINT  D$ ;" OPEN " ; F$ : PRINT  D$ ; " WRITE " ; F$
80  PRINT  C0 ;"  FOR  I  =  " ; A ; "  TO  " ; B
90  PRINT  C1 ;" READ  N : POKE  I , N "
```

```
100  PRINT C2 ; " NEXT I "
110  FOR I = A TO B
120  L = L + 1 : IF L = 40 THEN L = 1
130  IF L = 1 THEN PRINT : PRINT C ; " DATA " ; : C = C + D
140  PRINT PEEK ( I ) ;
150  IF I = B OR L = 39 THEN 170
160  PRINT " , " ;
170  NEXT I
180  PRINT : PRINT D$ ; " CLOSE " ; F$
190  HOME

] RUN
 START ADDRESS : 768

 END ADDRESS : 900

 FIRST LEBEL : 100

 LEBEL INTERVAL : 5

 FILE NAME : ABC

] NEW

] EXEC ABC

100  FOR I = 768 TO 900
105  READ N : POKE I , N
110  NEXT I
115  DATA 0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 1 6 , 17 , 18 , 19 , 20 , 21 , 22 ,23 ,
      24 , 25, 32 , 33 , 34 , 35 , 36 , 37 , 38 , 39 , 40 , 41 , 48 , 49 , 50 , 51 , 52 ,
     53 , 54 , 55 , 56
120  DATA  57 , 64 , 65 , 66 , 67 , 68 , 69 , 70 , 71 , 72 , 73 , 80 , 0 , 141 ,  0 ,
      3 , 104 , 205 , 0 , 224 , 240 , 3 , 173 , 128 , 192 , 96 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 ,
      0 , 0 , 0 , 0 , 0
125  DATA 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 255 , 255 , 239 , 255 , 239 , 255 , 239 ,
      255 , 239 , 255 , 255 , 255 , 255 , 255 , 255 , 255 , 255 , 255 , 239 , 255 ,
      239 , 255 , 0 , 1 6 , 239 , 255 , 2 , 3 , 255 , 4 , 5
130  DATA 6 , 255 , 255 , 239 , 255 , 239 , 255 , 7 , 8 , 239 , 255 , 247 , 9 ,
```

```
    10 , 11 , 12
] RUN
```

51. 如何将 BASIC 程序与机器语言程序混装

在编写程序时，常常需要将 BASIC 语言程序与机器语言程序相互配合来共同完成某项任务。这时的一般方法是将机器语言以名字 F 存入磁盘中，再在 BASIC 程序开头加一条调机器语言的语句:

```
5 PRINT CHR $ ( 4 ) "BLOAD F"
```

这种方法较繁琐，而且容易造成两个程序在内存中相互冲突。为此可采用以下几种混合装配的方法:

1. 使用 POKE 语句

在 BASIC 程序中使用 POKE 语句将机器语言子程序存入内存中。例如在 $ 0302 为超始地址的内存区域中有一段机器语言子程序:

```
0302- AD 30 C0 88 D0 05 CE 01
030A- 03 F0 09 CA D0 F5 AE 00
0312- 03 4C 02 03 60
```

可在 BASIC 程序中用 POKE 语句存放相应的内存区域中:

```
5   REM  PROGRAM  51.1
10  FOR  I  =  770  TO  790
20  READ  N : POKE  I , N
30  NEXT  I
40  DATA 173 , 48 , 192 , 136 , 208 , 65 , 206 , 1 , 3 , 240 , 9 , 202 , 208 , 245 , 17
    4 , 0 , 3 , 76 , 2 , 3 , 96
```

2. 使用监控命令

利用内存区域键盘缓冲区的特点,将监控状态下的赋值命令预先放入一个字符串中,需要时将它送入键盘缓冲区,然后调用监控程序中处理监控命令的子程序。

例如，上面的机器语言子程序,可用下面 BASIC 程序装入内存单元。

```
5   REM  PROGRAM  51.2
10  A$  =  " 302 : AD  30  C0  88  D0  05  CE  01  03  F0  09  CA  D0  F5  AE  00
    03  4C  02  03  60  N  D823G "
20  FOR  I  =  1  TO  LED ( A$ )
30  POKE  511  +  I , ASC ( MID  $  ( A  $  , I , 1 )  +  128  )
40  NEXT  I
```

50 POKE 72，0：CALL−144

3. 链接方法

将机器语言子程序紧接着 BASIC 程序结束符(连续的三个 0)存放，再将 BASIC 程序结束指针($ AF、 $ B0)进行修改，使之指向机器语言子程序结束处，让 BASIC 解释程序误认机器语言也是 BASIC 程序的一部分。因此，在使用 DOS 命令将 BASIC 程序存盘时，可将机器语言程序一起存入磁盘中，当使用 DOS 命令从磁盘中调取 BASIC 程序时，会将其后的机器语言程序一起调入内存单元中。

例如，有一 BASIC 程序的结束地址为 $ 1000(即 $ AF 内存单元中存入 $ 00， $ B0 内存单元中存入 $ 10)，机器语言程序占 $ 15 个内存单元，则将机器语言子程序依次存入 $ 1001—— $ 1015 内存单元中，并将 $ AF 内存单元中的数据改为 $ 15。

4. 插入方法:

利用 BASIC 程序的结构特点，可在程序的一开始(也可在程序的其它位置)建立一段"死区"(即不会执行到的区域)，然后将机器语言放入该死区中。例如:

```
5    GOTO  20
10   REM  1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0
20   ————————( 原用户程序 )
```

这时程序区内存单元中的数据为:

```
800—   00  09  08  05  00  AB  33  20
808—   00  23  08  0A  00  B2  31  32
810—   33  34  35  36  37  38  39  30
818—   31  32  33  34  35  36  37  38
820—   39  30  00  ……
```

这样内存区域 $ 80D—— $ 821 形成一个"死区"，可用来存放机器语言子程序。在装入机器语言程序时，一定要避免出现。

采用此方法后，机器语言能与 BASIC 程序同时存取，而且会在列程序时产生一些杂乱的内容，给人以错觉，因而有一定的保密效果。

52. 如何将机器语言程序存放在内存的安全区域

所谓内存的安全区域就是不会因输入、运行 BASIC 程序或引导 DOS 而破坏内存中原有数据的内存区域。常用的内存安全区域有:

1. 内存第三页的 $ 300− $ 38E 内存区域。

2. 在 BASIC 程序内存区域中制造一个安全区域（即"死区"，参看问题"如何将 BASIC 程序与机器语言程序混装"）。

3. 将机器语言程序存入 n－$9600 内存区域中，然后重新设定最高可用地址：

HIMEM：n

4. 在不使用 DOS 文件时，将机器语言子程序存入文件缓冲区：$9600－$9000 内存区域。

5. 将机器语言存入 DOS 程序中的一些未使用的内存区域：

$B6B3－$B6FC 共 73 个内存单元。

$BA69－$BA96 共 45 个内存单元。

$BC56－$BCFF 共 169 个内存单元。

53. 如何在 BASIC 程序中使用 DOS 命令

许多 DOS 命令都可以用在 BASIC 程序中，但采用如下的方法是不行的（以 CATALOG 为例）：

```
10 CATALOG
```

因为 DOS 命令无法用于延迟执行方式，这时计算机将 DOS 命令做变量处理。

为了使 BASIC 程序中能使用 DOS 命令，规定必须在 DOS 命令前加 PRINT CHR$（4），同时 DOS 命令要用引号括起来。例如：

```
10  D$ =CHR$ (4)
20  PRINT  D$ ;" CATALOG "
```

CHR$（4）实际上是 CTRL－D 的控制符（没有显示），因而也可以用下述方法在 BASIC 程序中使用 DOS 命令。

```
(1)    10  D$ =""        (引号内为无显示的 CTRL-D)
       20  PRINT  D$ ;"CATALOG"
(2)    10  PRINT " CATALOG " (引号内字母 C 前为无显示的 CTRL-D)
```

此外，应特别注意，使用 DOS 命令时，PRINT CHR$（4）前面必须刚刚执行完回车命令。

例如：

```
10  D$ = CHR $ (4)
20  PRINT "* *";
30  PRINT  D$ ;" CATALOG "
```

程序运行后，不会执行 DOS 命令 CATALOG，而是将它作为字符串打印出来。

解决上述问题的方法是在 CHR$（14）前面加回车符 CHR$（13）：

```
10  D$ = CHR$ (13) + CHR$ (4)
20  PRINT "* *";
30  PRINT  D$ ;" CATALOG "
```

在整数 BASIC 语言中，没有 CHR$ 函数，因此在 INTBASIC 程序中使用 DOS 命令，应采用在引号内键入 CTRL－D 键的方法。

54. 如何在 BASIC 程序中使用监控命令

中华学习机系统把内存区域的第 2 页（$200~$2FF）作为键盘缓冲区。在 BASIC 或监控状态下，用户可通过键盘对机器下各种命令或输入 BASIC 程序，键盘输入的字符的 ASCII 码在未按回车键前暂存在这一内存区域中。按回车键后，中华学习机对第二页内存区域进行扫描翻译，根据内存中 ASCII 码的内容作相应的处理。如果是带行号的 BASIC 程序，则将这些 ASCII 码移至内存的程序区，并改变相应的指针。如果是不带行号的命令，则立即按命令要求完成相应任务。再通过键盘输入新字符的 ASCII 码后，就将键盘缓冲区原 ASCII 码复盖了。这里需注意的是，在监控状态下存入键盘缓冲区的 ASCII 码是负 ASCII，在 FPBASIC 状态下存入键盘缓冲区的是正 ASCII 码和命令代码，在 INTBASIC 状态下存入的是负 ASCII 码和指令代码。

了解了键盘缓冲区的作用，就可以利用它。如果在键盘缓冲区存入有关命令的 ASCII 码，然后再调用扫描翻译的监控子程序，则中华学习机就能执行这些命令。根据此法，我们可以在 BASIC 程序中执行监控命令，在机器语言程序中执行 BASIC 命令和 DOS 命令。此处仅就在 BASIC 程序中执行监控命令举例加以分析。

例如，有时我们需要把高分辨率第一页绘好的图形转到第二页去显示（实际是将第一页内存单元中的图形数据移至第二页内存单元中），或将其它内存数据进行迁移。如果使用 BASIC 语言中的 POKE 和 PEEK 命令是可以完成的，但迁移的速度很慢。在监控状态下用 MOVE 命令却可以以很快的速度完成迁移工作。那么，是否可以在 BASIC 程序中使用监控状态的 MOVE 命令呢？采用下面程序中的方法是可以实现的。

```
5    REM  PROGRAM  54.1
10   DIM  X (15) , Y (15)
20   FOR  I =  1  TO  15
30   X  (I)  =  140  +  139  *  COS  ( 6.28  *  I / 15 ) : Y  ( I )  =  95  +  SIN
    ( 6.28  *  I  /  15 )
40   NEXT  I
50   HGR  :  POKE  49234 , 0 :  HCOLOR  =  7
60   FOR  I  =  1  TO  14 :  FOR  J  =  I  +  1  TO  15
70   HPLOT  X  ( I )  , Y  ( I )   TO  X  ( J )  , Y  ( J )
80   NEXT  J , I :  GET  A$
90   HGR2  :  GET  A$
100  S$  =  " 4000  <  2000 . 3FFFM  N  D823G "
110  FOR  I  =  1  TO  LEN   ( S  $  )
120  POKE  511  +  I  , ASC   (  MID$  ( S$  , I , 1 ))  +  128
130  NEXT  I :  POKE  72 , 0
140  CALL  -  144
150  GET  A$  :  TEXT  :  HOME
```

对程序做以下几点说明:

1. 程序的第 20 至第 80 语句是在高分辨率第一页绘制了如下所示的一个图形。

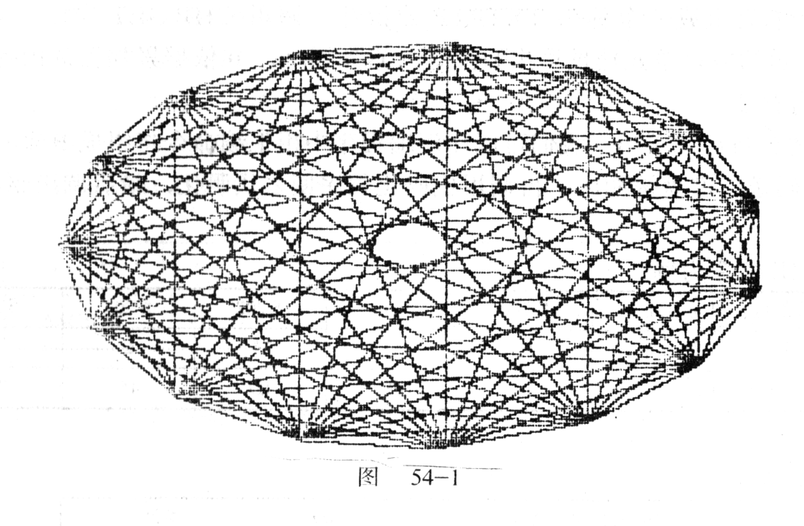


图 54—1

2. 第 90 语句是设定高分辨率第二页图形显示方式。

3. 第 100 语句是将监控的 MOVE 命令作为字符串赋给 S $。其中 $ 4000 是高分辨率图形第二页内存区域首地址; $ 2000 是第一页内存区域首地址, $ 3FFF 是第一页内存区域尾地址; M 是 MOVE 命令符号 (完成内存数据迁移工作); N 是监控状态下置正常显示方式的命令, 在此它起到间隔监控命令的作用; $ D832G 是运行 (命令符号为 G) 以 $ D832 为起始地址的子程序, 使机器从监控状态返回 BASIC 状态, 并继续执行以后的 BASIC 程序。

4. 第 110 至第 130 语句是将 S $ 中的监控命令, 依次送入键盘缓冲区, 模拟击键过程。其中加 128 是为了把正 ASCII 码变为相应的负 ASCII 码。第 130 语句是为使用监控中的扫描翻译子程序而给 CPU 标志位 P 置零。

5. 第 140 语句是调用监控中的扫描翻译子程序, 让机器执行键盘缓冲区中的监控命令, 完成图象数据迁移任务。

根据上述方法, 我们还可以在 BASIC 程序中使用其它监控命令。

55. 如何在汇编程序中使用 BASIC 命令

我们可以象在汇编语言程序中使用 DOS 命令那样, 在汇编语言程序中使用 BASIC 命令。这将给使用汇编语言编写应用程序带来极大的方便。

通常, 使用者通过键盘键入 BASIC 命令 (立即执行的 BASIC 命令, 例如 HOME), 计算机将键入的字符的 ASCII 码 (是正 ASCII 码) 送入以 $ 200 为起始地址的键盘缓冲区。然后, 计算机通过调用 CHRGET 取字符程序, 读入字符的第一个字符。若输入的是立即执行的 BASIC 命令, 则由代码化程序 (入口地址为 $ D559) 把该命令代

码化。其后，再用 CHRGET 程序从键盘缓冲区读入一个数据送至累加器 A，并进入驱动程序（入口地址为＄D828）进行解释执行。

取字符程序 CHRGET 存放在＄B1～＄C8 的内存单元中。在该程序中，由＄B9、＄B8 两个内存单元组成一个称为 TXTPTP 的指针。调用 CHRGET 程序，则将根据＄B9、＄B8 指示的内存单元地址取出数据送入累加器 A，并依据累加器的内容设置 CPU 的状态标志寄存器。

通过上述分析可以看出：我们可以在汇编程序中用汇编命令将组成 BASIC 命令的字符的 ASCII 码送入键盘缓冲区，再调用代码化程序使之代码化，最后调用驱动程序执行该命令。

以 HOME 命令为例：

存放的内存单元地址	＄200	＄201	＄202	＄203	＄204
字　　　符	H	O	M	E	
ASCII 码	48	4F	4D	45	00

代码化后为：

存放的内存单元地址	＄200	＄201
代　　码	97	00
命　　令	HOME	

在汇编程序中使用 HOME 命令的汇编程序为：

```
0300-  A9 48      LDA  #$48
0302-  8D 00 02   STA  $0200
0305-  A9 4F      LDA  #$4F
0307-  8D 01 02   STA  $0201
030A-  A9 4D      LDA  #$4D
030C-  8D 02 02   STA  $0202
030F-  A9 45      LDA  #$45
0311-  8D 03 02   STA  $0203
0314-  A9 00      LDA  #$00
0316-  8D 04 02   STA  $0204
0319-  A2 01      LDX  #$01
031B-  86 B9      STX  $B9
031D-  A2 FF      LDX  #$FF
031F-  86 B8      STX  $B8
0321-  20 B1 00   JSR  $00B1
0324-  20 59 D5   JSR  $D559
0327-  20 B1 00   JSR  $00B1
032A-  4C 28 D8   JMP  $D828
```

56. 如何在汇编语言程序中使用 DOS 命令

采用下面所述的方法，可以象在 BASIC 程序中使用 DOS 命令一样，在汇编语言程序中使用 DOS 命令。这将给使用汇编语言编写应用程序带来很大的方便。

通常立即执行 DOS 命令，是通过键盘将 DOS 命令（实质是组成 DOS 命令的字符的 ASCII 码）输入到以＄200 为起始地址的键盘缓冲区内。然后再调用 DOS 命令的扫描程序（入口地址为＄9FCD)，对 DOS 命令进行识别、判断和参数处理，并转入各个 DOS

命令的处理程序。

因此，我们可以采用下面的方法在汇编语言程序中使用 DOS 命令。其方法如下:

1. 首先用汇编语言将 DOS 命令的字符依次以 ASCII 码形式存入以 $ 200 为起始地址的单元（即键盘缓冲区）。输入完后，应送入一个 RETURN（回车符）的 ASCII 码。

2. 用 JMP $ 9FCD 命令调用 DOS 命令扫描程序（也可用 JSR $ 9FCD）。

例如，用汇编语言编写一个执行 CATALOG 命令的程序:

1. 将 CATALOG 及回车符的 ASCII 码用汇编语言送入以 $ 200 为起始地址的内存单元中（这里所用的 ASCII 码为负 ASCII 码）。

字 符	C	A	T	A	L	O	G	↵
ASCII 码	C3	C1	D4	C1	CC	CF	C7	8D

2. 在汇编语言程序末尾使用 JMP $ 9FCD 命令。

汇编语言程序如下:

```
0300-  A9 C3       LDA  #$C3
0302-  8D 00 02    STA  $0200
0305-  A9 C1       LDA  #$C1
0307-  8D 01 02    STA  $0201
030A-  A9 D4       LDA  #$D4
030C-  8D 02 02    SA   $0202
030F-  A9 C1       LDA  #$C1
0311-  8D 03 02    STA  $0203
0314-  A9 CC       LDA  #$CC
0316-  8D 04 02    STA  $0204
0319-  A9 CF       LDA  #$CF
031B-  8D 05 02    STA  $0205
031E-  A9 C7       LDA  #$C7
0320-  8D 06 02    STA  $0206
0323-  A9 8D       LDA  #$8D
0325-  8D 07 02    STA  $0207
0328-  4C CD 9F    JMP  $9FCD
032B-  00          BRK
```

57. 什么叫 BASIC 程序的复盖和链接

对于中华学习机和其它微型计算机，都存在一个内存小和程序大的矛盾。也就是说，实用的 BASIC 程序需要占据的内存空间往往大于计算机的实际内存容量。为了解决这个矛盾，需要采用 BASIC 程序的复盖和链接技术。

所谓 BASIC 程序的复盖和链接，就是将一个大的 BASIC 程序分为几个小的 BASIC 程序，并将这些小的 BASIC 程序存入磁盘中。使用时，将这些小 BASIC 自动地依次调入主机内存中。调入第一个小 BASIC 程序并运行它后，再调入下一个小 BASIC 程序并运行新调入的 BASIC 程序，按此执行下去直至运行完整的 BASIC 程序。

BASIC 程序在内存中存放时，形成两个区域，一个是程序区，一个是数据区。BASIC 程序的复盖是指先调入主机内存中的小程序运行结束后，不清除内存的程序区就调

入下一个小程序，复盖内存中的原程序后再运行之。所谓复盖就是前后二个小程序中相同行号的程序内容以新代旧，不同行号的程序行并存。BASIC 程序的链接是指先调入主机内存中的小程序运行结束后，首先清除内存的程序区，再调入下一个小程序。

BASIC 程序的复盖与链接就是否清除内存中的数据区又各分二种处理方式。清除内存数据区及其有关内存单元的处理方式，使各小程序的变量各自独立，相互没有联系；不清内存数据区及其有关内存单元的处理方式，使各小程序的变量具有联系，即前面小程序运行后的变量值是后面小程序变量的初值。这样，后面小程序可以利用前面小程序的运行结果。

归纳起来，将一个大程序分成若干小程序存入磁盘中，然后连续运行各小程序的处理方式有四种：

1. 保存内存程序区（复盖），保存内存数据区和有关内存单元。
2. 保存内存程序区（复盖），清除内存数据区和有关内存单元。
3. 清除内存程序区（链接），保存内存数据区和有关内存单元。
4. 清除内存程序区（链接），清除内存数据区和有关内存单元。

58. 如何实现 BASIC 程序的复盖和链接

一. BASIC 程序的链接

BASIC 程序的链接是指清除内存中的程序区后再从磁盘中调入新的程序到内存中，并运行这个新程序。它有两种处理方式。

1. 清除内存数据的链接

在一个程序中，用下面的语句代替程序中的语句 END。

格式：＜号行＞PRINNT CHR＄ (4)；“RUN＜下一个程序的名称＞”

例如，有两程序，其名称为 T-1 和 T-2，分别存入磁盘。

```
10  REM   T-1
20  FOR  I=1 TO 5
30  M=M+I
50  PRINT "M=";  M, "I=";  I
60  END
```

程序 T-2 是:

```
10  REM   T-2
20  FOR J = I+1 TO 10
30  M = M+J
40  NEXT  J
50  PRINT "M"; M, "I="; I, "J="; J
60  END
```

将程序 T−1 改为:

```
10  REM  T − 1
20  FOR   I = 1  TO  5
30  M + I
40  NEXT   I
50  PRINT  "M = ";  M,  "I = ";  I
60  PRINT   CHR $  (4) "RUN   T − 2"
```

将程序 T−1 调入内存并运行之，运行结果为:

```
M = 15          I = 6
M = 55          I = 0          J = 11
```

2. 保留内存数据区的链接

在前一个程序中，用下面的语句代替程序中的语句 END。

格式：<行号>PRINT CHR $ (4) "BLOAD CHAIN, A520": CALL 工 520 "<下面一个程序的名称>"

注意：(1) 必须将 DOS3.3 系统盘中的程序 CHAIN 拷贝到存放各 BASIC 程序所在的磁盘中。

(2) 代换的语句中，CALL520 与后面的引号之间不能有空格存在。

仍以上面的程序 T−1、T−2 为例。将程序 T−1 改为:

```
10  REM  T −1
20  FOR  I = 1  TO  5
30  M  =  M + 1
40  NEXT  1
50  PRINT  "M = ";  M,  "I = ";  I
60  PRINT   CHR $  (4) "BLOAD CHAIN,  A520":  CALL520 "T−2"
```

运行程序 T−1 后的结果是:

```
M = 15        I = 6
M = 49        I = 6        J = 11
```

对 INTBASIC 程序，可以更简单地进行保留内存数据区的链接。这时不必使用系统盘中的 CHAIN 程序，只需使用 INTBASIC 情况下的 DOS 命令 CHAIN 就可以完成链接任务。用下面语句代替程序中的语句 END。

格式：<行号>PRINT ""; "CHAIN<下一个程序名称>"

上面修改的语句中，PRINT 后面的双引号内为 CTRL−D。以上面的例子为例，程序 T−1 的第 60 语句应改为:

```
60  PRINT""; "CHAINT−2"
```

二. BASIC 程序的复盖

BASIC 程序的复盖是指保留内存中的程序区后再从磁盘中调入新的程序到内存中，并运行这个新程序。采用这种处理方式可以保留内存程序区的一些有用的程序行，使之与新调入的程序合并成最终的完整程序。BASIC 程序复盖也有两种处理方式。

1. 清除内存数据区的复盖

将后一个程序作为程序文件存入磁盘中，并在前一个程序中用下面的语句代替程序中的语句 END。

格式：<行号> PRINT CHR＄ (4)；"EXEC<下一个程序的程序行>"

例如，有两个程序，其名为 D–1 和 D–2。将程序 D–2 作为程序文件存入磁盘，名称定为 D2。

程序 D–1：

```
5   REM D-1
10  A$ = "$"
20 A = 1: B = 5: C = 1: B$ = A$
30  FOR I = A TO B STEP C
40 K = K+1: PRINT K; ": "A$;
50  FOR J = 1 TO 2 * I-1
60  PRINT TAB (20-I); K;
70  NEXT J
80  A$ = A$ +B$: PRINT
90  NEXT I
100  END
```

程序 D–2 为：

```
5    REM  D-2
10   A$  = "*"
20   A = 4: B = 1: C = -1: B$ = A$
100  END
```

将程序 D–2 后面增加一条语句：

```
63999 D$ = CHR$ (4): PRINT D$"OPEN D2": PRINT D$"WRITE D2": POCE 33, 30:
LIST 5, 100: PRINT "RUN": PRINT D$"CLOSE D2": POKE 33, 40: END
```

运行这条语句(RUN 63999)，产生 D2 的程序文件 D2。再在程序 D–1 中用下面的语句

代替语句 END：

```
100  PRINT  CHR$ (4); "EXEC D2"
```

运行修改后的程序 D–1，其运行结果如下：

```
]RUN
1:  $                          1
2:  $ $                       22
3:  $ $ $                    33333
4:  $ $ $ $                 4444444
5:  $ $ $ $ $             55555555555


1:  *                      1111111
2:  * *                     22222
3:  * * *                    333
4:  * * * *                   4
```

这时，内存中的程序（即复盖后的程序）为：

```
]LIST
   5  REM   D–2
   10 A$ = "*"
   20 A = 4:  B = 1:  C = 11:  B$ = A$
   30  FOR I = A TO B STEP C
   40 K = K + 1:  PRINT K; ":  "A$;
   50  FOR J = 1 TO 2 * I – 1
   60  PRINT TAB(20 – 1);  K;
   70  NEXT J
   80 A$ = A$ + B$:  PRINT
   90  NEXT I
   100  END
```

运行程序复盖时应注意以下几点：

(1) 程序复盖时应将内存中源程序的调程序文件的语句复盖掉，否则会引起死循环。

(2) 程序复盖处理后，DATA 语句的指针指在复盖后内存新程序第一条 DATA 语句的第一个数据。

(3) 在 EXEC 命令打开顺序文件后，如果文件中有 RUN 命令运行 BASIC 程序，程序中的 INPUT 语句将读入文件中下一个记录而不是等待键盘输入，如果下一个记录刚好是一个立即执行命令方式的 DOS 命令，则这个命令将在程序继续执行以前被执行，造成混乱。为了避免上述情况，文件中的 RUN 命令应放在最后。

4. 将＄801重新送入＄67和＄68内存单元中，即执行下述命令：

POKE 103，1：POKE104，8

这样就将第一个程序恢复了，使两个程序并接在一起。举例如下：

```
10  A$="$"
20  FOR I=1 TO 10
30  PRINT TAB (15-1); A$
40  A$=A$+"$ $"
50  NEXT I

]POKE 103, PEEK (175) -3
]POKE 104, PEEK (176)
]LIST

]LOAD T-2
]LIST

90  A$="*"
100  FOR I=1 TO 10
120  PRINT TAB (15-1); A$
130  A$=A$+"* *"
140  NEXT I
]POKE 103,1: POKE 104,8
]LIST
10  A$="$"
20  FOR I=1 TO 10
30  PRINT TAB (15-1): A$
40  A$=A$+"$ $"
50  NEXT I
90  A$="*"
100  FOR I=1 TO 10
120  PRINT TAB (15-1); A$
130  A$=A$+"* *"
140  NEXT I
```

60. 如何使用 & 命令

FPBASIC 语言中有一个功能很强的命令 & 。它允许用户通过 &命令无条件地由 FPBASIC 程序转至机器语言子程序。而且，还可以在 &命令后面设置一些参数，实现 FPBASIC 程序与机器语言子程序之间的参数传递。

当在 FPBASIC 程序中执行 &命令时，程序会无条件地转至地址为 $03F5 处的机器语言子程序去执行，其作用与执行 CALL1030 一样。由于以 $03F5 为起始地址的可供用户使用的内存单元较少，故用户可以在 $03F5 处存入一条 JMP **** 命令（其中 **** 为用户机器语言子程序入口地址），这样可使程序转至任意地址的机器语言子程序。

& 命令后面还可以带多个参数，中华学习机的 FPBASIC 解释程序提供了几个机器语言程序，可以用来处理 FPBASIC 程序与机器语言子程序之间传递的参数。各机器语言处理子程序的入口地址及其功能如下:

1. 入口地址为 $DD67 的机器语言子程序可以读出 &命令后面的输入参数（常量、变量或表达式的值），将其按一定存放格式存入 $9D——$A3 内存单元（叫 FPBASIC 的浮点累加器）。

2. 入口地址为 $E6FB 的机器语言子程序可以将 $9D——$A3 浮点累加器内存放的浮点数转换为一个单字节的整数，放入 X 寄存器。若该数小于 0 或大于 255，则显示"?ILLEGAL QUANTITY ERROR"错误信息，并返回 FPBASIC 状态。若该数含小数，则舍去小数点后面的数。

3. 入口地址为 $E10C 的机器语言子程序可以将浮点累加器中的浮点数转换为一个两字节的整型数，并放入 $A0 与 $A1 内存单元中（高位在前，低位在后）。数值范围为-32768——32767，若数值超出范围，则显示" ?ILLEGAL QUANTITY ERROR "错误信息，并返回 FPBASIC 状态。

4. 入口地址为 $DFE3 的机器语言子程序可以读出 &后面输入的参数变量，将其存放的内存单元首址的低位存入 A 寄存器，高位存入 Y 寄存器。

5. 入口地址为 $DEBF 的机器语言子程序可以检验读入的是否是逗号。如果不是，则显示"?SYNTANERROR "错误信息，并返回 FPBASIC 状态。这样 &命令后可输入多个由逗号分开来的参数。

6. 入口地址为 $E746 的机器语言子程序可以对 &命令后的两个以逗号分隔的算术表达式进行处理，将第一个算术表达式的值送入 $50 内存单元，第二个算术表达式的值送入 Y 寄存器。

7. 入口地址为 $E2F2 的机器语言子程序可以将由寄存器 A (高位) 和 Y (低位)组成的整数转换为浮点数，存入 $AD——$A3 浮点累加器中。

在使用 &命令时常用到的零页内存单元有:

$B8.$B9:　存放程序要读取的字符或指令代码所在内存单元的地址。

$81.82:　存放目前使用的变量名称。

& 命令的应用举例如下:

二. 使用 EXEC 命令

使用 EXEC 命令将两个程序并接的方法如下:

1. 将程序 2 由键盘输入或由磁盘调入主机内存，注意将其行号与程序 1 行号错开。

2. 添加下列程序:

```
63999 D$ = CHR$ (4): PRINT D$ "OPEN LISTING": PRINT D$
"WRITE LISTING": POKE 33, 30: LIST 0, 63998: PRINT D$ "CLOSE": END
```

运行内存中的这段程序 (RUN 63999)，即可将程序 2 作为顺序文件 LISTING 存贮在磁盘内。

3. 用 LOAD 命令或由键盘将程序 1 送入主机内存中。

4. 键入 EXEC LISTING 命令，即可完成两个程序并接的任务。

举例如下:

```
90  A$ ="*"
100  FOR I=1 TO 10
120  PRINT TAB (15-1); A$
130  A$ =A$+"* *"
140  NEXT I
63999 D$ =CHR$ (4): PRINT D$ "OPEN LISTING": PRINT D$ "WRITE LIST-
ING": POKE 33, 30: LIST 0, 63998: PRINT D$ "CLOSE": END
]RUN
                    *
                  * * *
                * * * * *
              * * * * * * *
            * * * * * * * * *
          * * * * * * * * * * *
        * * * * * * * * * * * * *
      * * * * * * * * * * * * * * *
    * * * * * * * * * * * * * * * * *
  * * * * * * * * * * * * * * * * * * *

]LOAD T-1
]LIST
10  A$ ="$"
20  FOR I=1 TO 10
30  PRINT TAB (15-1); A$
40  A$ =A$+"$ $"
```

```
50  NEXT I

]EXEC LISTING
]LIST
10  A$ ="*"
20  FOR I=1 TO 10
30  PRINT TAB (15-1); A$
40  A$ =A$+"$ $"
50  NEXT I
90  A$ ="*"
100  FOR I=1 TO 10
120  PRINT TAB (15-1); A$
130  A$ =A$+"* *"
140  NEXT I
]RUN
```

```
          $                               *
         $ $ $                           * * *
        $ $ $ $ $                       * * * * *
       $ $ $ $ $ $ $                   * * * * * * *
      $ $ $ $ $ $ $ $ $               * * * * * * * * *
     $ $ $ $ $ $ $ $ $ $ $           * * * * * * * * * * *
    $ $ $ $ $ $ $ $ $ $ $ $ $       * * * * * * * * * * * * *
   $ $ $ $ $ $ $ $ $ $ $ $ $ $ $   * * * * * * * * * * * * * * *
  $ $ $ $ $ $ $ $ $ $ $ $ $ $ $ $ $   * * * * * * * * * * * * * * * * *
 $ $ $ $ $ $ $ $ $ $ $ $ $ $ $ $ $ $ $   * * * * * * * * * * * * * * * * * * *
```

三. 改变向量指针

这种方法比较简单，只需改变 BASIC 内部向量指针。其方法如下：

1. 将一个 BASIC 程序输入到机器内存中，程序从地址为 $ 800 的内存单元开始存放。这时，$ 67、$ 68 存放程序首地址，$ 67 内存中存放低八位，$ 68 内存中存放高八位，即（$ 67）=1，（$ 68）=8；而 $ AF、$ B0 内存单元中存放程序尾地址，$ AF 内存中存放低八位，$ B0 内存中存放高八位。

2. 将 $ AF、$ 80 内存单元存放的程序尾地址减了后，存入 $ 67、$ 68 内存单元中，即执行如下命令：

POKE 103，PEEK（175）-3

POKE 104，PEEK（176）

完成上述命令后，程序 1 被屏蔽了，不会因从磁盘中调入程序 2 而受破坏。

3. 从磁盘调入程序 2。要求程序 2 的最小行号应大于程序 1 的最大行号。

2. 保留数据区的复盖

保留数据区的复盖，实际是先执行清除数据区的复盖得到复盖的新程序，再执行保留数据区的链接。处理过程举例如下（仍以上述的程序 D−1、D−2 为例）:

(1) 将上述程序 D−2 与 D−1 复盖后的程序以名字 D−3 存入磁盘。

(2) 将程序 D−1 中 END 语句改为:

```
100  PRINT  CHR$ (4)"BLOAD CHINA, A520":  CALL 520"−3"
```

然后，运行修改后的程序 D−1，其运行结果为:

```
]RUN

1:  $                          1
2:  $ $                        22
3:  $ $ $                    33333
4:  $ $ $ $                 4444444
5:  $ $ $ $ $             55555555555

6:  *                       6666666
7:  * *                      66666
8:  * * *                     888
9:  * * * *                    9
```

59. 如何将两个 BASIC 程序并接在一起

将两上 BASIC 程序并接在一起有三种方法。

一. 使用 RENUMBER 程序

将 DOS3 · 3 系统主盘中的 RENUMBER 程序调入内存后，就可以将两个 BASIC 程序并接起来。其方法是:

1. 用 LOAD 命令从磁盘中调入程序 1 或通过键盘输入程序 1。

2. 键入&H 命令，将主机内存中的程序 1 保护起来。

3、将磁盘中的程序 2 调入主机内存中。

4. 使用&H 命令＜起始行号＞，1＜行号间隔＞命令，将程序 2 的行号与程序 1 的行号错开（即使程序 2 的最小行号大于程序 1 的行号）。

5. 键入&M 命令。即可将两个程序按行号次序并接起来。用 LIST 命令即可观察到。

举例如下:

```
]LOAD T−1
```

```
]LIST
10  A$ ="$"
20  FOR 1=1 TO 10
30  PRINT TAB (15-1); A$
40  A$ =A$+"$ $"
50  NEXT I

]&H
PROGRAM ON HOLD, USE"&H"TO RECOVER
]LOAD T-2
]LIST
5  A$ ="*"
10  FOR I=1 TO 10
20  PRINT TAB (15-1); A$
30  A$ =A$+"* *"
40  NEXT I

]&F100, I10
]LIST
100  A$ ="*"
110  FOR I=1 TO 10
120  PRINT TAB (15-1); A$
130  A$ =A$+"* *"
140  NEXT I

]&M
]LIST
10  A$ ="$"
20  FOR I=1 TO 10
30  PRINT TAB (15-1); A$
40  A$ =A$+"$ $"
50  NEXT I
100  A$ ="*"
110  FOR I=1 TO 10
120  PRINT TAB (15-1); A$
130  A$ =A$+"* *"
140  NEXT I
```

例 1. 用 &命令代替有关的 BASIC 命令或 DOS 命令

就象前面提到的，在 $ 03F5——$ 03F7 内存单元存入一条 JMP * * * * 命令(* * * * 是 BASIC 或 DOS 命令的入口地址)，即可用 &命令代替某一 BASIC 命令(或保留字)或 DOS 命令。

例如：用&命令代替 CATALOG 命令

```
*3F5: 4C 6E A5
```

即存入一条 JMP $ A56 命令。

DOS 命令入口地址见表 60-1 。

表 60-1 DOS 命令的入口地址 (十六进制表示)

DOS 命令	入口地址	DOS 命令	入口地址	DOS 命令	入口地址
PR#	A299	APPEND	A298	WRITE	A510
IN#	A22E	OPEN	A2A3	READ	A51B
MON	A233	CLOSE	A2EA	INIT	A54F
NOMON	A23D	BDAVE	A331	CATALOG	A56E
MAXFILES	A251	BLOAD	A35D	FP	A57A
DELETE	A263	BRUN	A38E	INT	A59E
LOCK	A271	SAVE	A397	EXEC	A5C6
UNLOCK	A275	LOAD	A413	POSITION	A5DD
VERIFY	A27D	RUN	A4D1		
RENAME	A281	CHAIN	A4F0		

例 2. 创造发音的 BASIC 命令

用机器语言编写一个唱歌的发音子程序，存入 $ 300——$ 317 内存区域中，$ 50. $ 51 分别存放音调与节拍。用 &命令带音调与节拍两个数据去调用发音机器语言子程序。

发音机器语言子程序如下：

```
*300L
0300- 20 46 E7    JSR   $E74
0303- 86 51       STX   $51
0305- AD 30 C0    LDA   $C030
0308- 88          DEY
0309- D0 04       BNE   $030F
030B- C6 51       DEC   $51
030D- F0 08       BEQ   $0317
030F- CA          DEX
0310- D0 F6       BNE   $0308
0312- A6 50       LDX   $50
0314- 4C 05 03    JMP   $0305
0317- 60          RTS
```

用 &命令调此子程序时，需要在 $ 3F5——$ 3F7 内存单元中存入 JMP $ 300 命令，即键入：

```
* 3F5: 4C 00 03
```

下面一个使用 &命令反复演奏 “Do Rei Mi ”歌曲的 BASIC 程序：

```
10  FOR I = 1 TO 5
20  & 96 + ( INT ( RND (1) * 3) + 3) * 96 / 5,(I - 2 * INT
  (I / 2) + 1) * 60
30  NEXT I
40  GOTO 10
```

例 3. 创造新 GOTO 命令

FPBASIC 语言中的 GOTO 语句，要求 GOTO 保留字后面必须是正整数行号值，不允许是变量或表达式。这里用 &命令调用一个小机器语言程序，可使 &命令代替 GOTO 命令，而且允许其命令后面为变量或表达式。

机器语言程序如下:

```
0300-  20  7B  DD   JSR   $DD7B
0303-  20  52  E7   JSR   $E752
0306-  20  1A  D6   JSR   $D61A
0309-  90  03       BCC   $030E
030B-  4C  41  D9   JMP   $D941
030E-  A2  5A       LDX   #$5A
0310-  4C  12  D4   JMP   $D412
```

键入

```
*3F5 : 4C 00 03
```

应用程序如下:

```
10   INPUT "A=";A
20   & A
30   GOTO 10
100  PRINT "100 LINE"
110  GOTO 10
200  PRINT "200 LINE"
210  GOTO 10
```

```
RUN
A=  100
100 LINE
A=  200
200 LINE
A=
```

将 20 语句改为:

```
20 & A * 100
```

当输入 1 或 2 后，程序会转至 100 或 200 语句执行。

例 4. 扩充多个新 BASIC 命令

接着例 3 中的机器语言程序，输入下面这段机器语言程序，可建立三个新 BASIC 命令

&G <表示>　　: 新 GOTO 命令

&l　　: 列程序清单

&H　　: 清文本屏幕

机器语言程序如下:

如此下去，可将一个程序一步步执行完毕。

十七. 跟踪执行命令

＊ <地址>T

从指定地址处开始执行机器语言指令，每执行完一条指令，就将指令的汇编指令及其机器语言代码，以及各寄存器的值显示出来，直至执行完一条 BRK 指令后才中断。

63. 如何使用小汇编命令

使用机器语言编写程序，需要记住 151 个操作码，这是很困难的。用汇编语言编写程序较容易，但输入计算机时需借助一块能编辑和汇编的软盘，掌握这个编辑与汇编软件的使用很麻烦。故可以应用主机 ROM 中的小汇编程序 (MINI-ASSEMBLER)来输入汇编程序。使用方法如下:

一. 进入小汇编状态

首先进入监控状态，再进入小汇编状态:

```
] CALL-151
 * D350G
!
```

如果是由整数 BASIC 状态进入监控状态的，应:

```
>CALL-151
 * F666G
!
```

二. 往规定的内存单元中送入汇编符号指令

```
! 300 :LDA   #$10
```

屏幕显示出

```
300-A9   10      LDA   #$10
```

再输入第二条指令:

```
!  STA $2000
```

屏幕显示出:

```
302 - 8D  00  20      STA    $2000
```

注意: (1) 操作数与操作地址用十六进制数， $ 可省去。

(2) 不可使用符号地址。

(3) 条件转移指令中的地址码要用转移目标处的绝对地址，不能用转移步长或符号地址。

三. 列出源程序和对应的机器码

```
! $0300 L
300 - A9   10          LDA #  $10
302 - 8D   00   20     STA      $2000
!
```

四. 运行汇编程序

! $ 0300 G

五. 检查内存单元的数据

例如，检查 $ 300 内存单元中的数据

! $ 0300

六. 修改内存单元中的数据

! $ 0300 :FF FF FF FF

七. 由小汇编状态退出

如退回原监控状态，则键入:

! $ D360G

或 ! $ FF69G

如退回整数 BASIC 状态，则键入:

! INT

在小汇编状态下，可以使用各种监控命令，使用方法是在各监控命令前面加一个“ $ ”符号。注意在中文状态下不宜使用小汇编程序。

64. 如何修改机器语言程序

输入一段机器语言到内存区域中后，如发现有错误，可采用以下方法来修改。

一. 修改某一内存单元中的数据

例如，修改程序一中 $ 305 内存单元中的数据，使之为 $ 85。

程序一:

0300 – 18 A9 D0 A0 00 80 07 84

0308 – 06 00 B1 06 60

* 305 :85

二. 修改内存单元中一段数据

例如，将 $ 305—— $ 308 内存单元中的数据改为: A9、00、20、A2。

* 305 :A9 00 20 A2

三. 删除某段内存单元的数据

例如，删除 $ 305—— $ 308 内存单元的数据。

* 305 < 309.30CM

四. 插入数据

例如在 $ 305～ $ 308 处插入数据: A9 00 20 A2 。

1. 首先将 $ 305～ $ 30C 中的数据迁移至某一内存区域中:

* 6000 < 300.30CM

2. 键入要插入的数据

* 305:A9 00 20 A2

3. 将迁移走的程序移至 $ 309 为起始地址的内存单元中:

* 309 < 6000.600CM

其中， addr 是算术表达式，其值为要调用的机器语言子程序的入口地址。

通过 CALL 命令可以实现在 FPBASIC 程序中调用机器语言子程序。如果在 CALL 命令地址数据 addr 后面加上参数:

CALL addr ,A,A⋯

可将 A 、B 等参数传送给以 addr 值为入口地址的机器语言子程序。在用户机器语言子程序中加入 JMP ＄DEBE 或 JMP ＄DFE3 等命令，即可完成相应的参数接收、处理工作。

62. 如何使用监控命令

监控程序固化在主机的 20 M 中，开机后便可使用。

一. 监控状态的进入与退出

从 BASIC 状态进入监控状态:

]CALL－151

*

从监控状态退回 BASIC 状态:

* 3D0G (或 CTRL－C. CTRL－RESET 等)

]

二. 检查内存单元中的数据

1. * <地址>： 显示指定地址的内存单元的数据。

2. *.<地址>： 显示从当前地址到给定地址所有内存单元的数据。

3. * <地址 1>·<地址 2>： 显示从地址 1 到地址 2 的所有内存单元的数据。

4. * <地址>

   *： 显示后续地址尾数为 7 或 F 的数据（最多是八个数据）

三. 修改内存单元中的数据

1. * <地址>： <数据>： 把给定的数据写入给定地址的内存单元中。

2. * <地址>： <数据 1> <数据 2>⋯<数据 n>： 把 n 个数据依次送入当前地址开始的几个内存单元中。

四. 将内存中一段内存单元的数据迁移到另一段内存单元中

* <地址 1> < <地址 2>· <地址 3> M： 把地址 2 到地址 3 的内容迁移到从地址 1 开始的内存区域中。

五. 比较两段内存单元中的数据

* <地址 1><<地址 2>·<地址 3>: 比较从地址 2 到地址 3 的数据是否与从地址 1 开始的内存单元的数据一样。不一样时，显示出不同的数据和两个内存单元的相应地址。一样时什么也不显示。

六. 运行机器语言程序

* <地址> G ： 运行起始地址为给定地址的机器语言程序。

七. 列汇编程序和相应的机器码

* <地址> L ： 从给定地址起列各条指令及相应的机器码（列 20 行）。

* L　：从当前行开始到各条指令及相应的机器码（列 20 行）。

八. 检查和改变各寄存器数据

* CTRL−E　:显示 A.X.Y.P.S 各寄存器数据。

* <数据 1>　<数据 2>…　<数据 n>：将数据依次送入 X，…寄存器中，个数不能大于 5。

九. 将内存区域中的数据存入磁带

* <首地址>·<末地址> W：将内存区域中从首地址到末地址的内存单元中的数据存入磁带。

十. 将磁带的内容读入主机内存中

* <首地址>.<末地址>R　：将磁带中的一段内容读入给定的内存区域中。

十一. 改变显示方式

*I　：设屏幕字符为反转显示。

* N　：设屏幕字符为正常显示。

十二. 强迫使监控程序转至 $3F8。

十三. 接通与关闭外设

* <槽口号> CTRL−P　：接通与给定槽口号相接口卡及外设（P 取值为 0—7）。例如:

* 1 CTRL−P　：如果槽口接打印机卡与打印机，槽口号定为 1，则这时接通打印机。

* 6 CTRL−P　：接通驱动器。

* 0 CTRL−P　：接通显示器。

该命令不能在中文状态下使用。

十四. 十六进制加、减法

* <数 1>+<数 2>：求十六进制数之和。

* <数 1>−<数 2>：求十六进制数之差。

十五. 多重命令

允许在同一命令行内写入多个以空格分开的监控命令，只要字符个数小于 245 个就可以。但若在命令中使用修改存储器内容的命令，则在该命令之后，应写一个单字母的命令(通常用 N 命令)，用于把随后的命令隔开。在多重命令中，单一字母命令之间不需用空格分隔。例如:

* <地址>LLL:　可连续列三个 20 行汇编指令及相应的机器代码。

* <地址 1> L <地址 2> G：先列出地址 1 开始的 20 条汇编指令及相应的机器代码，然后运行地址 2 开始的机器语言程序。

十六. 单步执行命令

* <地址>S:执行指定地址处的一条机器语言指令，并将该指令的汇编指令及相应的机器代码显示出来，同时还显示执行完指令后各寄存器的值。如果想继续执行下一条机器语言指令，可再键入:

*S

```
0313- A0 00      LDY  #$00          0321-          JMP  $0300
0315- B1 B8      LDA  ($B8), Y      0324- C9 4C    CMP  #$4C
0317- 48         PHA                0326- D0 03    BNE  $032B
0318- C8         INY                0328- 4C A5 D6 JMP  $D6A5
0319- 20 98 D9   JSR  $D998         032B- C9 48    CMP  #$48
031C- 68         PLA                032D- D0 03    BNE  $0332
031D- C9 47      CMP  #$47          032F- 4C 58 FC JMP  $FC58
031F- D0 03      BNE  $0324         0332- 4C C9 DE JMP  $DEC9
```

键入:

```
* 3F5: 4C 13 03
```

应用程序如下:

```
10  INPUT "A=";A
20  & GA * 100
30  GOTO 10
100  PRINT "100 LINE"
110  & H
120  & L
130  END
200  PRINT "200 LINE"
210  GOTO 10
```

61. 如何实现 BASIC 与机器语言子程序间的参数传递

为了充分发挥 FPBASIC 编程简单、调式方便和机器语言运行速度快的优点，程序员常采用 FPBASIC 语言与机器语言结合的方法来编写程序。这就存在如何实现 FPBASIC 与机器语言子程序间参数传递的问题。下面介绍三种实现参数传递的方法。

一. 用 &命令

执行 &命令，会使程序无条件地转至地址为 $ 3F5 去执行机器语言程序，同时还可以将 &后面的参数传递到机器语言程序中，将机器语言程序中的参数传递给 FPBASIC 程序。处理参数传递要利用 FPBASIC 解释程序中的几个机器语言子程序。有关 &命令的使用和处理参数传递的机器语言子程序的介绍可参看“如何使用 &命令”这一问题。

二. 用 USR 函数

USR 函数的使用格式为：

USR(算术表达式)

用户通过 USR 函数可以调用机器语言子程序，并向子程序提供一个参数，同时还可以将机器语言子程序处理完的结果返回 FPBASIC 程序。例如，执行 N=USR (算术表达

式)，可以将算术表达式的值送至 ＄9D——＄A3 浮点累加器，然后转至 ＄0A 处，通过 ＄0A——＄0C 处的 JMP ＊＊＊＊命令，跳转到以入口地址为 ＊＊＊＊的机器语言子程序去执行。当执行 RTS 命令返回 FPBASIC 程序时，USR 函数自动将 ＄9A——＄A3浮点累加器中的数传送给变量 N 。

将存于浮点累加器中的算术表达式的值转换为一字节或两字节的整数，可使用入口地址为 ＄E6FB 或 ＄E10C 的机器语言子程序。将用户机器语言子程序的结果转换为浮点数存于＄9A——＄A3 浮点累加器，可使用入口地址为 ＄E2F2 的机器语言子程序。

与 &命令类似，如果要调用多个用户机器语言子程序可用 POKE 语句改变 ＄0B. ＄0C 中的入口地址数据。

用 USR 函数传递的参数可以是多个，其格式是

USR(算术表达式)，A， B…

它可以将 A.B 等参数传给用户机器语言子程序，而用户子程序可以通过调用入口地址为＄DEBE .＄DFE3 等 FPBASIC 解释程序中的机器语言子程序来接收处理这些参数。

例如，下面的程序可以完成 2＊INT(N) 的计算(−16383＜＝N＜＝16383)。

用户的 BASIC 程序及运行结果为:

```
10  POKE 10,76: POKE 11,0: POKE 12,3
20  INPUT "N=";N
30 S = USR (N)
40  PRINT "S=";S
50  GOTO 20
]RUN
N=23.1452
S=46
N=6543.23
S=13086
```

用户的机器语言子程序为:

```
0300- 20 0C E1        JSR   $E10C
0303- 06 A0           ASL   $A0
0305- A5 A0           LDA   $A0
0307- 06 A1           ASL   $A1
0309- 69 00           ADC   #$00
030B- A4 A1           LDY   $A1
030D- 20 F2 E2        JSR   $E2F2
0310- 60              RST
```

三. 用 CALL 命令

CALL 命令的格式是:

CALL addr

65. 如何显示标志寄存器的八个标志位数据

在调试机器语言程序时，常常需要了解标志寄存器各位的数据，这时可在程序适当位置加一条运行下面这个机器语言子程序的命令（JMP ＄0300 或 JSR ＄0300)即可。

能显示标志寄存器八个标志位数据的机器语言子程序如下：

```
*300.348
0300- D8 20 8E FD A9 D0 20 ED
0308- FD A9 BD 20 ED FD A5 48
0310- 20 DA FD 20 48 F9 A2 00
0318- A5 48 48 BD 42 03 20 ED
0320- FD 68 0A 48 2A 29 01 09
0328- B0 20 ED FD A9 A0 20 ED
0330- FD E8 E0 07 B0 0A E0 02
0338- D0 E1 68 0A 48 4C 1B 03
0340- 68 60 CE D6 C2 C4 C9 DA
0348- C3
```

```
*300G
P = 33    N0 V0 B1 D0 I0 Z1 C1
*48:4
*300G
P=04      N0 V0 B0 D0 I1 Z0 C0
```

三、打印机的使用

66. 如何使用打印机面板上的按钮与指示灯以及进行自检

打印机面板上通常有三个按钮和四个指示灯。三个按钮可以单独使用也可以组合使用，使用方法是：

1. ON LINE 按钮：其作用是控制打印机是否与主机接通。开机后，打印机自动与主机接通，这时 ON LINE 按钮的指示灯亮。当第一次按下该键后，会使打印机与主机脱离，同时它的指示熄灭。交替按该按钮，会使打印机在与主机接通或断开两种状态下变换，指示灯也会做相应的指示。在打印机发生故障(如打印纸用完)时，打印机会自动置为与主机脱离的状态，ON LINE 指示灯自动熄灭。排除故障后，按下 ON LINE 按钮，打印机又恢复到与主机接通的状态。例如，接上打印纸后，按此按钮，打印机 ON LINE 指示灯亮。同时继续打印刚才没打印完的信息。

2. LF 按钮和 FF 按钮：接通电源后，当打印机处于与主机脱离状态时，按一下 LF 按钮可使打印纸向前走一行，按一下 FF 按钮可使打印纸向前走一页。

如果将 LF、FF 按钮同时按下，再打开电源开关，则打印机采用十六进制码打印。

如果将 LF 按钮按下，同时打开电源开关，则打印机自动进行自检工作，即将打印机所能打印的字符顺序打印出来。

四个指示灯的作用是：

(1) POWER：表示打印机接通电源。

(2) READY：表示打印机已准备接收信息。

(3) PAPER OUT：表示打印纸已接近走完或打印机发生故障。

(4) ON LINE：表示打印机是否与主机接通，灯亮表示接通，灯灭表示断开。

打印机具有自检功能，它可以检查打印头、色带、马达及走纸等工作情况和和打印质量。其操作方法是：按下 LF 按钮的同时打开电源开关，则打印机会自动将可能打印的所有字符顺序打印出来。通过打印情况可检查打印机的质量。

67. 如何使用打印机内部的小开关

打开打印机面板上的小盖板或打开打印机盖，其内有两组小开关，第一组 SW1 有八个小开关，第二组 SW2 有四个小开关，其作用如表 67−1 至 67−3 所示(对 EPSON FX 型打印机而言)：

表 67-1　SW1 开关的功能与条件

小开关	功　能	ON	OFF	出厂状态
SW1-1	每行字符数	132	80	OFF
SW1-2	数字 0 的表示法	0	0	OFF
SW1-3	打印纸打完检测	不用	用	OFF
SW1-4	输入缓冲器	用	不用	OFF
SW1-5	开电源后的打印方式	粗体	常体	OFF
SW1-6	内部字符组	见表 67-2		
SW1-7	置不同西文字符			
SW1-8				

表 67-2 内部字符组

开关＼国家	美国	英国	法国	德国	丹麦	瑞典	意大利	西班牙
SW1-6	ON	ON	ON	ON	OFF	OFF	OFF	OFF
SW1-7	ON	OFF	ON	OFF	ON	ON	OFF	OFF
SW1-8	ON	OFF	OFF	ON	ON	OFF	ON	OFF

表 67-3 SW2 开关的功能与条件

小开关	功能	ON	OFF	出厂状态
SW2-1	SLCTIN 信号内部情况	固定	不固定	ON
SW2-2	报警	用	不用	ON
SW2-3	打印至页末剩下一行时跳新页	用	不用	OFF
SW2-4	换行	用回车命令自动换行	由主机换行	

注：(1)使用汉字时，需将 SW2-4 小开关置 ON 位置。(2) 小开关 SW2-1 的含义是：当它处于 ON 状态时，表示打印机永远处于“选中”方式，外信号不能退出它；当开关处于 OFF 状态时，可以由外部软件码来“选中”或“退出选中”。

68. 如何使打印机每行打印的字符个数增加

对于 80 列打印机，开机后它设置为每行打印 40 个字符。如果要使打印机每行打印的字符个数增加，可键入：

POKE　1656+n, m

其中 n 为打印机卡所插的槽口号，一般为 1。m 是一行打印的字符个数。例如，要一行打印 70 个字符，则键入：

POKE　1657, 70

要恢复到每行打印 40 个字符，可键入：

POKE　1657, 40

69. 如何打印字符的上标或下标

在打印字符的上标或下标前使用下述命令可打印出字符的上标或下标：打印标准型字符上标的命令是：

PRINT CHR $ (27)+CHR $ (83)+CHR $ (0);

打印标准型字符下标的命令是：

PRINT CHR $ (27)+CHR $ (83)+CHR $ (1);

从打印标准型字符上标或下标回到打印标准型字符的命令是：

PRINT CHR $ (27)+CHR $ (84);

举例如下：

```
5  REM PROGRAM 69.1
10  D $ = CHR $ (4): PRIN D $ ″PR#1″
20  PRINT ″A″;
30  PRINT  CHR $ (27) + CHR $ (83) + CHR $ (0);″100″
35  PRINT  CHR $ (27) + CHR $ (84)
40  PRINT ″B″;
50  PRINT  CHR $ (27) + CHR $ (83) + CHR $ (1);″LOAD″
60  PRINT  CHR $ (27) + CHR $ (84)
70  PRINT  D $ ″PR#0″
80  END
]RUN
```

A^{100}

B_{LOAD}

70. 如何打印放大或缩小的字符

在要打印的字符前使用下述命令可使字符放大或缩小打印出来：

1. 打印标准型的宽字符

打印标准型宽字符的命令是：

PRINT CHR $ (14);

从打印标准型宽字符回到打印标准型字符的命令是：

PRINT CHR $ (20);

2. 打印缩小的字符

打印缩小的字符的命令是：

PRINT CHR $ (15);

从打印缩小的字符回到打印标准型字符的命令是：

PRINT CHR $ (18);

3. 打印缩小型宽字符

打印缩小型宽字符的命令是:

PRINT CHR $ (15)+CHR $ (14);

从打印缩小型宽字符回到打印标准型字符的命令是:

PRINT CHR $ (18);

举例如下:

```
5   REM  PROGRAM 70.1
10  D $ = CHR $ (4): PRINT D $ "PR#1"
20  PRINT  CHR $ (14);: GOSUB 100
30  PRINT  CHR $ (20)
40  PRINT  CHR $ (15);: GOSUB 100
50  PRINT  CHR $ (18)
60  PRINT  CHR $ (15)+ CHR $ (14);: GOSUB 100
70  PRINT  CHR $ (18);: GOSUB 100
80  PRINT  D $ "PR#0"
90  END
100  PRINT "ABCDEFGH1234"
110  RETURN
]RUN
ABCDEFGH1234
ABCDEFGH1234
ABCDEFGH1234
ABCDEFGH1234
```

71. 如何打印斜体字

在要打印的字符前使用下面的命令可使打印机打印出斜体字符:

PRINT CHR $ (27)+CHR $ (52);

从打印斜体字符回到打印标准型字符的命令是:

PRINT CHR $ (27)+CHR $ (53);

举例如下:

```
5   REM  PROGRAM 71.1
10  D $ = CHR $ (4): PRINT D $ "PR#1"
20  PRINT  CHR $ (27) + CHR $ (52);
30  PRINT "ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890"
40  PRINT  CHR $ (27) + CHR $ (53);
50  PRINT "ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890"
60  PRINT  D $ "PR#0"
]RUN
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890
```

72. 如何使打印的字符变黑变浓

在要打印的字符前使用下述命令，可使打印的字符变黑变浓。

1.打印加强型的字符

所谓加强型的字符，是指组成字符的点在水平方向增加了一倍，如图 72–1 所示。

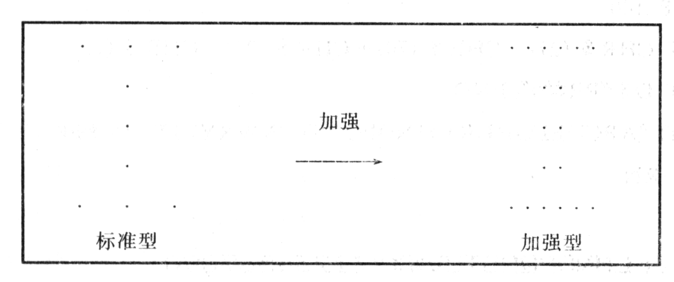


图 72–1　加强型字符

打印加强型字符的命令是:

```
PRINT CHR $ (27)+CHR $ (69);
```

从打印加强型字符回到打印标准型字符的命令是:

```
PRINT CHR $ (27)+CHR $ (70);
```

2.打印重叠型字符

所谓重叠型字符，是指每个字符打印两次，两次打印的字符在垂直方向约有 0.1mm 的偏移。

打印重叠型字符的命令是:

```
PRINT  CHR $ (27)+CHR $ (71);
```

从打印重叠型字符回到打印标准型字符的命令是:

```
PRINT  CHR $ (27)+CHR $ (72);
```

3. 打印重叠的加强型字符

打印重叠的加强型字符的命令是:

```
PRINT  CHR $ (27)+CHR $ (69)+CHR $ (27)+CHR $ (71);
```

从打印重叠的加强型字符回到打印标准型字符的命令是:

```
PRINT  CHR $ (27)+CHR $ (70)+CHR $ (27)+CHR $ (72);
```

举例如下:

```
5  REM  PROGRAM 72.1
```

```
10  D$ = CHR$ (4): PRINT D$"PR#1"
20  PRINT  CHR$ (27)+CHR$ (69);: GOSUB 100
30  PRINT  CHR$ (27)+CHR$ (70)
40  PRINT  CHR$ (27)+CHR$ (71);: GOSUB 100
50  PRINT  CHR$ (27)+CHR$ (72)
60  PRINT  CHR$ (27)+CHR$ (69)+CHR$ (27)+CHR$ (71);
70  GOSUB 100
80  PRINT CHR$ (27)+CHR$ (70)+CHR$ (27)+CHR$ (72);
90  PRINT D$"PR#0": END
100  PRINT "ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890"
110  RETURN
] RUN
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890

ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890

ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890
```

73. 如何改变页长和页间隔

在打印程序或运行结果时，常常希望将程序打印成一页一页的形式，以便装订。打印机接通电源（也未对它下过有关命令）后，自动设置每页长为66行，间隔为0。改变页长和页间隔的命令如下：

1. 设每页长为n行

   PRINT CHR$(27)+CHR$(67)+CHR$(n)

2. 设每页长为m英寸

   PRINT CHR$(27)+CHR(67)+CHR$(0)+CHR$(m)

3. 设页间隔为n行

   PRINT CHR$(27)+CHR$(78)+CHR$(n)

4. 设页间隔为0页

   PRINT CHR$(27)+CHR$(79)

5. 使打印机向前空一页

   PRINT CHR$ 12)

6. 使打印机向前空走一行

   PRINT CHR$(10)

命令的作用同按一下LF键一样，使打印纸进一行，打印头不一定会移到左边。

举例如下：

```
3  REM PROGRAM 73.1
5  PRINT  CHR$ (4)"PR#1"
10  PRINT  CHR$ (27)+CHR$ (67)+CHR$ (10)
```

```
20  PRINT  CHR $ (27)+CHR $ (78)+CHR $ (4)
30  FOR I = 1 TO 17
40  PRINT "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
50  NEXT I
60  PRINT  CHR $ (27)+CHR $ (67)+CHR $ (66)
70  PRINT  CHR $ (27)+CHR $ (79)
80  PRINT  CHR $ (4)"PR#0"
] RUN

ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ

ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ

ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ

ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

74. 如何打印重叠的字符和使打印机的扬声器发声

一. 打印重叠的字符

灵活使用下述命令可以打印重叠的字符。

1. 使打印头向左退一格。

PRINT CHR $ (8)

2. 使打印头移回左边。

PRINT CHR $ (13)

对于 EPSON 打印机，在出厂时就调整为自动换行，也就是说，在使用 PRINT

CHR $ (B)命令时，它不但执行将打印头移回左边，而且还使打印纸向前进一行，即自动换行。如果要取消自动换行，可键入下述命令:

    POKE 1529, 255

如果要恢复自动换行功能，可键入:

    POKE 1529, 0

二.使打印机的扬声器发声

使用下面这个命令可使打印机的扬声器发声:

    PRINT CHR $ (7)

举例如下:

例 1.

```
5  REM PROGRAM 74.1
10  D$ = CHR$ (7)
20  PRINT D$"PR#1"
30  FOR I = 1 TO 5 :A$ =A$ +CHR$(124) : B$ + CHR$ (8) : NEXT I
40  PRINT "OOOOO";B$ ; A$
50  PRINT "OOOOO";B$ ; "------"
60  PRINT D$"PR#0"
] RUN
ΦΦΦΦΦ
ΘΘΘΘΘ
```

例 2.

```
5  REM PROGRAM 74.2
10  D$ = CHR$ (7)
20· PRINT D$;"PR#1"
30  POKE 1529,255
40  PRINT "OOOOOOOOOOOO";
50  PRINT CHR$ (13);
60  FOR I = 1 TO 12
70  PRINT CHR$ (124);
80  NEXT I
90  PRINT : POKE 1529,0
100  PRINT D$;"PR#0"
] RUN
ΦΦΦΦΦΦΦΦΦΦΦΦ
```

75. 如何确定打印行的间距

接通打印机电源后，打印机自动将行间距设置为 1／6 英寸。如果要改变行间距可使用以下命令:

1. 设行间距为 1／8 英寸的命令是:

   PRINT CHR $ (27)+CHR $ (0);

2. 设行间距为 7/72 英寸的命令是:

   PRINT CHR $ (27)+CHR $ (1);

3. 设行间距为 1／6 英寸的命令是:

   PRINT CHR $ (27)+CHR $ (2)

4. 设行间距为 n／72 英寸的命令是:

   PRINT CHR $ (27)+CHR $ (65)+CHR $ (n)

其中 n 是自然数, 且 1⩽n⩽85

5. 设行间距为 n／216 英寸的命令是:

   PRINT CHR $ (27)+CHR $ (3)+CHR $ (n);

举例如下:

```
5   REM  PROGRAM 75.1
10  D $  = CHR $  (4): PRINT D $ "PR#1"
20  FOR N = 3 TO  21  STEP  3
30  PRINT CHR $  (27) +CHR $  (65)+ CHR $  (N);
40  FOR I = 1  TO  3
50  PRINT "LINE SPACING";N;"／72 INCH"
60  NEXT  I
70  PRINT   CHR $  (27) + CHR $  (65)+CHR $  (12) +CHR $  (13)
80  NEXT   N
90  PRINT  D $ "PR#0"
] RUN
LINESPACING3/72INCH
LINESPACING3/72INCH
LINESPACING3/72INCH

LINESPACING6/72INCH
LINESPACING6/72INCH
LINESPACING6/72INCH

LINESPACING9/72INCH
LINESPACING9/72INCH
LINESPACING9/72INCH

LINESPACING12/72INCH
LINESPACING12/72INCH
LINESPACING12/72INCH

LINESPACING15/72INCH
LINESPACING15/72INCH
LINESPACING15/72INCH

LINESPACING18/72INCH
LINESPACING18/72INCH
LINESPACING18/72INCH

LINESPACING21/72INCH
LINESPACING21/72INCH
LINESPACING21/72INCH
```

76. 如何打印字符下方的底线

在要划底线的字符被打印之前使用下面的命令:

PRINT CHR $ (27)+"____"+CHR $ (1);

然后再在划底线的字符刚打印完的后面使用下面的命令:

PRINT CHR $ (27)+"____"+CHR $ (0);

举例如下:

```
5  REM  PROGRAM 76.1
10  D$ = CHR$ (4); PRINT D$"PR#1"
20  PRINT "ABC";
30  PRINT  CHR$ (27)+"-"; CHR$ (1);
40  PRINT  "DEFG";
50  PRINT  CHR$ (27)+"-"+CHR$ (0);
60  PRINT "HIJKL"
70  PRINT  D$"PR#0": END
] RUN

] ABCDEFGHIJKL
```

77. 如何打印英文小写字母

EPSON 打印机可以将英文大写字母转换成英文小写字母打印出来。实现转换的方法是在打印小写字母前执行一次。

PRINT CHR $ (23);

命令。如果要回到打印英文大写字母状态,可再执行一次上述命令。

举例如下:

```
5  REM  PROGRAM  77.1
10  D$ = CHR$ (4): PRINT  D$"PR#1"
20  PRINT "T";
30  PRINT  CHR$ (23);"HIS  BOOK"; CHR$ (23);
40  PRINT " CAN NOT ";
50  PRINT  CHR$ (23);"BE COPIED  BY  ANY  MEANS."; CHR$ (23)
60  PRINT  D$"PR#0": END
70  PRINT  D$"PR#0": END
] RUN
This book CAN NOT be copied by any means.
```

78. 如何控制打印字符的左起位置和右边终止位置

在要打印的字符前执行下述命令，可使字符从第 N 个字符位处开始打印:

PRINT CHR $ (27);"l";CHR $ (n);

其中，l 为英文 L 的小写字母。

举例如下:

```
5   REM  PROGRAM  78.1
10  D $  =  CHR $ (4): PRINT D $ "PR#1"
20  PRINT "1234567890123456789012345678901234567890123456789O"
30  PRINT  CHR $ (27);"1"; CHR $ (10)
40  PRINT "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
50  PRINT  CHR $ (27);"1"; CHR $ (0);
60  PRINT "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
70  PRINT  D $ "PR#0"
80  END
] RUN

12345678901234567890123456789012345678901234567890
          ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZX
```

如果要使打印机一行打印的字符个数不超过 m 个，可在要打印的内容前执行下述命令:

PRINT CHR $ (27);"Q";CHR $ (M);

举例如下:

```
5   REM  PROGRAM 78.2
10  D $  =  CHR $ (4): PRINT D $ "PR#1"
20  PRINT "12345678901234567890123456789012345678901234567890"
30  PRINT  CHR $ (27);"Q"; CHR $ (13)
40  PRINT "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
50  PRINT  CHR $ (27);"Q"; CHR $ (80)
60  PRINT "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
70  PRINT  D $ "PR#0"
80  END
] RUN
```

```
1234567890123456789012345678901234567890123456789 0

ABCDEFGHIJKLM
NOPQRSTUVWXYZ

ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

综合使用上述两个命令，可以同时控制要打印内容的左起位置和一行打印字符的个数，打印的字符个数为 m−n 个。

举例如下:

```
5  REM  PROGRAM 78.3
10  D$ =  CHR$ (4): PRINT D$"PR#1"
20  PRINT "12345678901234567890123456789012345678901234567890"
30  PRINT  CHR$ (27);"1"; CHR$ (10)
35  PRINT  CHR$ (27);"Q"; CHR$ (24)
40  PRINT "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
50  PRINT  CHR$ (27);"1"; CHR$ (0);
55  PRINT  CHR$ (27);"Q"; CHR$ (80);
60  PRINT "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
70  PRINT D$"PR#0"
80  END
] RUN
12345678901234567890123456789012345678901234567890

ABCDEFGHIJKLMN
OPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

79. 如何打印高分辨率图形

利用中华学习机监控程序中的一个程序可以在 EPSON 打印机上打印高分辨率图形。具体方法如下(设打印机卡插在 1 号槽):

一. 执行 PR#1 命令，使主机与打印机接通。

二. 往地址为 1913 的内存单元中存放图形打印数据 N($1 \leqslant N \leqslant 225$)

$$N = P+L+I+D+U$$

其中 P 为必选项，其它的各项可不选用。各项的含义如下:

1. P 为高分辨率图形页的选择项

P = 1 时表示打印高分辨率第一页图形。

P = 2 时表示打印高分辨率第二页图形。

P＝3 时表示并排打印高分辨率第一、二页图形。

2. L 是决定打印两页高分辨率图形的逻辑操作项

L＝4 表示与(AND)操作，即当第一、二页高分辨率图形在屏幕上相同位置均有色点睦才打印。

L＝8 表示或(OR)操作，即当第一、二页高分辨率图形在屏幕上相同位置只要有一页图形有色点就打印。

L＝6 表示异或(EOR)操作，即当第一、二页高分辨率图形在屏幕上相同位置二页同时有色点或无色点则不打印，而其中只有一页有色点则打印。

3. I 决定图形颜色是否反相

I＝0 表示用黑色打印图形而底色为白色。

I＝32 表示底色为黑色而图形为白色。

4. D 决定图形是否放大打印

D＝0 表示图形不放大打印。

D＝64 表示将图形放大一倍打印。

5. U 决定图象整幅或行打印

U＝0 表示图形整幅打印。

U＝128 表示将图形对应的文本显示方式下的某一行(在高分辨率图形显示方式下为八行色点)打印出来可以用 VTAB 语句来确定要打印的行(1～24 行)。这里要注意的是：当连续打印多行时，打印出的各行图形间会有 4／74 英寸的间隔。

例如，打印放大的第二页高分辨率图形可键入：

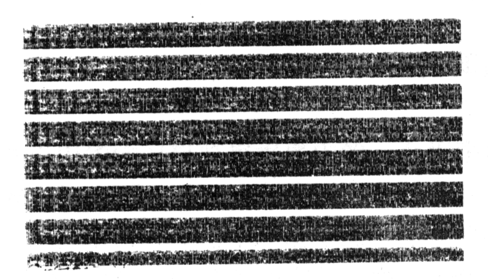
POKE 1913, 66

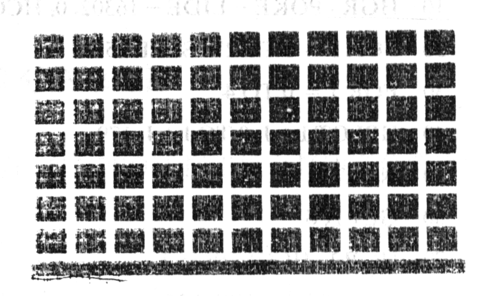
执行或操作打印两页图形可键入：　　　POKE 1913, 9 或 POKE 1913, 10

三. 键入 CTRL−Q，不用按回车键，打印机就自动将图形打印出来。

以下是将高分辨率第一页的横线图案和第二页纵线图案进行几种打印的程序和打印结果：

```
10   HGR : POKE - LIDE - 16302, 0: HCOLOR = 3
20   FOR I = 0 TO 279 STEP 25
30   FOR J = 0 TO 4
40   HPLOT I + J, 0 TO I + J, 179
50   NEXT J, I
60   GET A $
70   HGR2 : HCOLOR = 3
80   I = 0 TO 191 STEP 25
90   FOR J = 0 TO 4
100  HPLOT 0, I + J TO 279, I + J
110  NEXT J,I
120  GET A $
130  POKE - 16299,0: POKE - 163 00,0: GOTO 130
```





80. 如何实现文本屏幕的硬拷贝

一. BASIC 语言方法

运行下面的 BASIC 程序可实现文本屏幕硬拷贝:

```
5  REM PROGRAM 80.1
10  PR# 1: POKE 1657,80
20  FOR Y = 1 TO 24
30  FOR X = 0 TO 39
40  A = SCRN(X,Y * 2-1) * 16+SCRN(X,Y * 2-2)
50  PRINT CHR$ (A);
60  NEXT X: PRINT : NEXT Y
```

二. 机器语言方法

运行下面的机器语言程序可实现文本屏幕的硬拷贝:

```
* 300.34C
0300- 8A 48 98 48 A2 00 A0 00    0328- 90 C0 C8 C0 28 D0 F1 A9
0308- 8A 48 4A 29 03 09 04 85    0330- 8D 2C C1 C1 30 FB 8D 90
0310- 29 68 29 18 90 02 69 7F    0338- C0 A9 8A 2C C1 C1 30 FB
0318- 85 28 0A 0A 05 28 85 28    0340- 8D 90 C0 E8 E0 18 D0 BE
0320- B1 28 2C C1 C1 30 FB 8D    0348- 68 AA 68 A8 60
```

81. 如何打印低分辨率图形

中华学习机通常情况下只能打印高分辨率图形，而无法打印低分辨率图形。打印低分辨率图形。需先将低分辨率图形转换成高分辨率图形，然后将转换成高分辨率的图形打印出来。

一. BASIC 语言方法

运行下面的 BASIC 语言程序，屏幕会显示出转换成高分辨率图形的过程。当转换完毕后，屏幕显示“TO PRINT?”，问用户是否打印。如果打印则按“Y”键。

BASIC 程序如下:

```
5999  REM PROGRAM 81.1
6000  LOMEM: 16384: HGR
6005  FOR I = 0 TO 39
6010  FOR J = 0 TO 39
6015  SC = SCRN(I,J)
6020  IF SC > 0 THEN GOSUB 6050
6025  NEXT J,I
6030  INPUT " TO PRINT ";Y $
```

```
6035  IF Y$ < > "Y" THEN 6045
6040  PR#1: PRINT CHR$ (17): PR#0
6045  END
6050  IF SC = 4 OR SC = 12 THEN HC = 1: GOTO 6075
6055  IF SC = 3 OR SC = 1 THEN HC = 2: GOTO 6075
6060  IF SC = 8 OR SC = 9 OR SC = 11 OR SC = 13 THEN HC = 3
  :   GOTO 6075
6065  IF SC = 6 OR SC = 7 OR SC = 2 THEN HC = 5: GOTO 6075
6070  HC = 1
6075  HCOLOR = HC
6080  FOR K = 0 TO 3
6085  HPLOT I * 7,J * 4+K TO (I+1) * 7-1,J * 4+K
6090  NEXT K
6095  RETURN
```

82. 如何使打印机连续打印磁盘中多个程序的清单

有时我们需要将磁盘中的许多程序的清单依次打印出来。这时，如果用键盘输入 LOAD 和 LIST 命令的方法，会花费人们很多时间。如果我们将要通过键盘下的各项命令建立一个顺序文件，然后用 EXEC 命令顺序执行文件中的各项命令，即可实现打印机连续自动打印磁盘中许多程序清单的任务。

例如，磁盘中有三个程序，其名字分别为 D-1、D-2 和 D-3。用下面这个程序可以建立一个连续打印这几个程序清单的命令文件“LISTING”然后用 EXEC 命令执行“LISTING”文件中的各项命令，完成连续打印的任务。

程序如下:

```
5   REM PROGRAM 82.1
10  D$ = CHR$ (4)
20  PRINT D$;"OPEN LISTING"
30  PRINT D$;"WRITE LISTING"
40  PRINT "PR#1"
50  FOR I = 1 TO 3
60  PRINT "LOAD D-";I
70  PRINT "LIST"
80  NEXT I
90  PRINT "PR#0"
100  PRINT D$;"CLOSE LISTING"
```

用 EXEC 命令执行 LISTING 文件中各项命令后，打印机连续打印的结果为:

```
]EXEC LISTING
```

```
10 PRINT"ABCDEFG"
20 PRINT"12345"

10 PRINT"* * * * * * * * *"
20 PRINT"* * * * * * * * *"

10 PRINT"$ $ $ $ $ $ $"
20 PRINT"%%%%%%%%"
```

83. 如何打印汉字

使用中华学习机在汉字系统下打印汉字，要求所使用的打印机须与 EPSON MX-80 Ⅲ型打印机相兼容，如 FX-80Ⅲ、CP-80Ⅲ、FX-100 和 YAMATO 等型号的打印机。而且要求打印的接口为 CENTRONICS 接口，打印机必须定义为 I/O 槽口的一号槽。

进入汉字系统后，可以使用以下 BASIC 命令来控制和打印汉字。

1. 设置打印方式

格式：POKE1659,n

其中 n=1——15,可置 15 种字型进行汉字打印。n=0 为不打印，进入汉字系统后，打印方式自动置 0。

2. 设置字间距

格式：POKE1787,n

其中 n=0——255，可设字间距在 0——255 个点之间。进入汉字系统后，字间距自动置为 1。所谓的字间距是指所打印的 ASCII 字符间的点距离，一个汉字占用二个 ASCII 字符的位置。

3. 设置行距

格式：POKE2043,n

其中 n 为打印行之间的点间距，n 取值范围为 0—255 之间，即行间距可在 0—255 个点之间。进入汉字系统后，行间距自动置 1。

4. 设置行允许数

格式：POKE2043,n

其中 n 为一行所允许打印的汉字个数，n 取值范围为 0——255 之间。进入汉字系统后，行允许字数自动置为 40。

由于一行所允许打印的汉字个数与打印机本身的缓存区大小有关，一般 80 列打印机的缓存区为 400 个字节左右，所以一行允许打印的汉字个数的定义与所选择的打印方式以及设定的字间距有关。

在汉字状态下打印程序清单，除了要设定汉字字型、字间距，行间距与每行打印字数（后三项可省略）外，还应加上 LIST 命令，并将它们写成一行程序加在需要打印的源程序的首行。例如：

```
1  POKE 1659 ,6: POKE 1915 ,3:POKE 1787 ,2:POKE 2043,30:LIST
10       .
         .       (源程序)
         .
```

运行程序，即可打印出 6 号字型，字间距为 2，行间距为 3，一行 30 个汉字的程序清单。

如果要使用 MX-80Ⅱ、FX-80Ⅱ、RX-80Ⅱ等Ⅱ型机打印汉字，用户必须要修改打印控制命令。

1. 修改走纸命令

在Ⅱ型打印机上没有 n／216 英寸走纸功能，所以必须把打印驱动程序中使用的 ESC3m 命令修改成 ESC Am 命令，这样才能打印汉字。该命令序列放在打印驱动程序的 LFCODEO、LFCODE2、LFCODE4（控制上半行打印走纸）和 LFCODE3（控制下半行打印走纸）中。

在进入汉字系统后，我们可以执行下面二段程序之一，便能在 MX-80Ⅱ型打印机上打印汉字。

程序一： 1 POKE 916,27 :POKE 918,65 :POKE 919,8:POKE 920,8

程序二：

```
1000-  A9  1B  8D  94  03  A9  41  8D
1008-  96  03  A9  08  8D  97  03  8D
1010-  98  03  60
```

在执行上述程序后,前四种打印字型不能使用，执行 POKE 1659,n 命令时，n 取值范围为 5—15 之间。

2. 修改打印密度

打印密度控制的命令格式为 ESC Xn1n2,这里 X 为"K"时表示标准密度打印，X 为"L"时表示倍密度打印。在汉字打印驱动程序中，我们运用的是倍密打印。如果用户希望使用标准密度打印，可通过对 TPCODE 单元进行修改来实现。在进入汉字系统后，只要在用户程序中执行 POKE 915,75 命令，便可实现打印密度的转换。

对走纸命令与打印密度修改后，当机器返回文本显示状态，再重新进入汉字状态，只要不破坏 $ 47B(1147)单元的数据，修改的值不会因显示状态的改变而遭到破坏。

四、中华学习机图形显示

84. 中华学习机有哪几种显示方式

中华学习机有以下十种显示方式:

1. 文本第一页显示。
2. 文本第二页显示。
3. 高分辨率图形全屏幕第一页显示。
4. 高分辨率图形全屏幕第二页显示。
5. 高分辨率图形与文本混合第一页显示。
6. 高分辨率图形与文本混合第二页显示。
7. 低分辨率图形全屏幕一页显示。
8. 低分辨率图形全屏幕第二页显示。
9. 低分辨率图形与文本混合第一页显示。
10. 低分辨率图形与文本混合第二页显示。
11. 汉字显示。

所谓文本显示，即屏幕可显示 24 行 * 40 列字符，高分辨率图形显示是屏幕可显示 192 行 * 280 列色点，低分辨率图形显示是屏幕可显示 48 行 * 40 列色块，图形与文本图混合显示是指屏幕分成上下两个区域，所谓第一、第二页显示，是指可以绘两幅画面(字符或图形)，各幅画面的数据在内存中占据不同的区域。

如果用 T 表示文本显示，P 表示图形显示，H 表示高分辨率显示，L 表示低分辨率显示，A 表示图形与文本混合显示，B 表示全屏幕图形显示，1 表示第一页显示，2 表示第二页显示，则十种显示方式可表示为:

T−1	P−H−A−2
T−2	P−L−B−1
P−H−B−1	P−L−B−2
P−H−B−2	P−L−A−1
P−H−A−1	P−L−A−2

为了使中华学习机屏幕为某种显示方式，可利用屏幕显示软开关。将相应的内存单元中送入数值“0”，即可将屏幕设定为相应的显示方式。屏幕软开关内存元地址及其功能见表 84−1。

表 84-1 屏幕显示软开关内存单元地址及功能

	内存单元地址			功能
	正十进制地址	负十进制地址	十六进制地址	
1	49232	−16304	$ C050	P
	49233	−16303	$ C051	T
2	49234	−16302	$ C052	B
	49235	−16301	$ C053	A
3	49236	−16300	$ C054	1
	49237	−16299	$ C055	2
4	49238	−16298	$ C056	L
	49239	−16297	$ C057	H

例如，设定 P−H−B−2 显示方式，可运行以下程序:

```
10  POKE −16304,0:POKE −16297,0
20  POKE −16302,0:POKE −16299,0
```

在 FPBASIC 语言中，可用 TEXT 命令设定 T−1 显示方式，用 GR 命令设定 P−L−A 显示方式，它不设定几页，如果在使用 GR 命令前屏幕显示第一页，则使用 GR 命令后屏幕处于 P−L−A−1 显示方式，用 HGR2 命令设定 P−H−B−2 显示方式。在使用 GR、HGR 和 HGR2 命令时，还将绘图区清为黑色。

中文显示使用了高分辨率第二页，屏幕可以显示 17×10 个汉字或 34×10 个字符。直接按下“中文”键即可由文本状态进入汉字状态，直接按下“西文”键可以返回文本状态。

85. 如何实现不显示的高分辨率绘图

所谓不显示绘图，即在屏幕上看不见绘图过程的绘图。中华学习机设置了高分辨率绘图页软开关内存单元(地址为 230)，往该内存单元送入某数据后，即可设定计算机在相应高分辨率绘图页上绘图。其数据为:

POKE 230,32 (在第一页绘图)

POKE 230,64 (在第二页绘图)

POKE 230,96 (在第三页绘图)

POKE 230,128 (在第四页绘图，不能直接显示)

POKE 230,160 (在第五页绘图，不能直接显示)

计算机在某页绘图不一定显示这一页图形，因此巧妙地应用屏幕显示软开关和绘图页软开关可以 实现不显示绘图。例如，可以让屏幕显示第一页，而在第二上绘图；或让屏幕处于文本显示方式，而进行高分辨率绘图。下面就是屏幕处于文本显示方式，而计算机进行高分辨率绘图(绘棋盘格)的程序。

```
5  REM PROGRAM 85.1
10  HGR2 : HCOLOR= 3
20  TEXT : HOME
30  FOR I = 1024 TO 2039
40  POKE I,170
50  NEXT I
60  POKE 230,64
70  FOR X =  0 TO 270 STEP 5
80  HPLOT X,0 TO X,180
90  NEXT X
100  FOR Y = 0 TO 180 STEP 5
110  HPLOT 0,Y TO 270,Y
120  NEXT Y
130  GET A$
140  POKE  - 16304,0: POKE  - 16297,0: POKE  - 16299,0: POKE  -16302,0
```

设定绘图页后，如要清绘图页内存单元，可使用 CALL 62450 命令。

86. 如何使高分辨率图形显示屏幕的底色改变

如果要将高分辨率图形显示屏幕清为某种颜色(即改变底色)，可在设定该颜色并绘一个点后，使用 CALL 62454 命令。例如，将高分辨率图形显示屏幕清为红色。程序如下:

```
10  HGR2:HCOLOR=5:HPLOT 1,1
20  CALL 62454
```

87. 如何快速清除高分辨率图形

使用 HGR 或 HGR2 清除高分辨率图形的速度较慢，为了提高清除高分辨率图形的速度，可采用下述方法:

1. 键入下述机器语言:

```
0300-  A6 08      LDX   $08
0302-  A9 20      LDA   #$20
0304-  85 08      STA   $08
0306-  E0 01      CPX   #$01
0308-  F0 04      BEQ   $030E
030A-  A9 40      LDA   #$40
030C-  85 08      STA   $08
030E-  A5 08      LDA   $08
0317-  A9 00      LDA   #$00
0319-  85 07      STA   $07
031B-  91 07      STA   ($07),Y
031D-  C8         INY
031E-  D0 FB      BNE   $031B
0320-  A6 08      LDX   $08
0322-  E4 09      CPX   $09
0324-  F0 05      BEQ   $032B
```

```
0310- 18        CLC                 0326- E6 08      INC   $08
0311- 69 1F     ADC  #$1F           0328- 4C 1B 03   JMP   $031B
0313- 85 09     STA  $09            032B- 60         RTS
0315- A0 00     LDY  #$00
```

```
*300.32B
0300-  A6  08  A9  20  85  08  E0  01
0308-  F0  04  A9  40  85  08  A5  08
0310-  18  69  1F  85  09  A0  00  A9
0318-  00  85  07  91  07  C8  D0  FB
0320-  A6  08  E4  09  F0  05  E6  08
0328-  4C  1B  03  60
```

2. 如果是清高分辨率第一页图形，可在程序中适当位置加入下述语句：

<行号> POKE 8,1:CALL 768

如果是清高分辨率第二页图形可在程序中适当位置加入下述语句：

<行号> POKE 8,2:CALL 768

举例如下：

```
5  REM PROGRAM 87.1
10  D$ = CHR$(4)
20  PRINT D$;"BLOAD HGR"
30  PRINT D$;"BLOAD HPLOT-5"
40  POKE -16304,0: POKE -16297,0
50  GET A$
60  POKE 8,1: CALL 768
```

程序中，HGR 为上述机器语言存入磁盘中的文件名，HPLOT-5 为磁盘中某一图形信息的文件名。

88. 如何给文本显示屏幕开设窗口

文本窗中是指文本显示方式下屏幕显示字符的区域。开机后，文本窗口设定为最大状态，即可以显示 24 行×40 列字符。如果将有关数据存入 $20～$23 四个内存单元中，则可以设定文本窗口的大小和位置。有关数据参看表 88-1 和图 88-1。

表 88-1　窗口的内存单元地址和窗口数据

内存单元地址	功　　能	正常值	最小值	最大值
$20 (32)	窗口左边与最大值显示区左边的间距	0	0	39
$21 (33)	窗口宽度，即窗口内一行的字符个数	40	1	40
$22 (34)	窗口顶部距最大显示区域上方的间距	0	0	24
$23 (35)	窗口底部距最大显示区域上方的间距	24	0	24

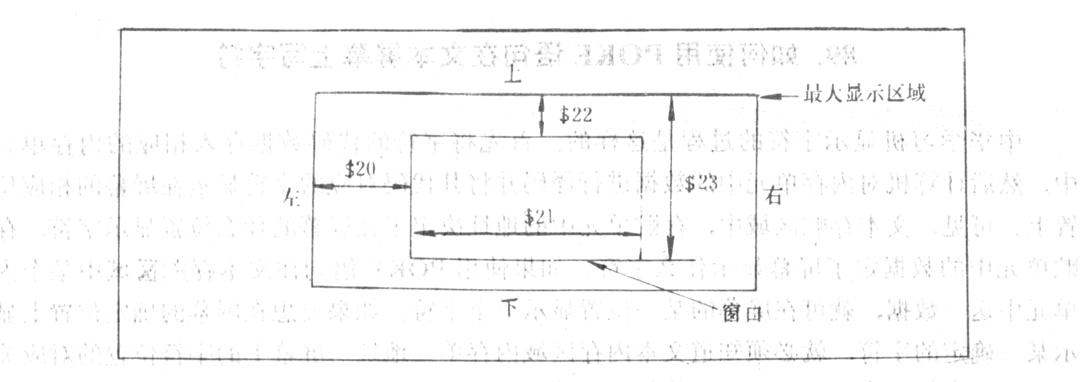


图 88-1　窗口示意图

设定窗口和在窗口内显示字符应注意以下几点:

1. $20 与 $21 内存单元中的数值和应不大于 40。

2. $22 内存单元中的数不大于 $23 内存单元中的数。

3. 若要在窗口内打印字符，需将光标移至窗口内。HOME 命令清窗口字符，并将光标移至窗口左上角。

4. VTAB 语句的使用与窗口设定无关，而 HTAB 语句和 TAB 函数的使用与窗口设置有关。

5. 表 88-1 中所说的间距是指能显示的字符个数。

例如，开设一个如图 88-2 所示的窗口，其背景为 24 行"*"，窗口内清为黑色，在窗口内第二行第 10 列显示一个"$"字符。程序如下:

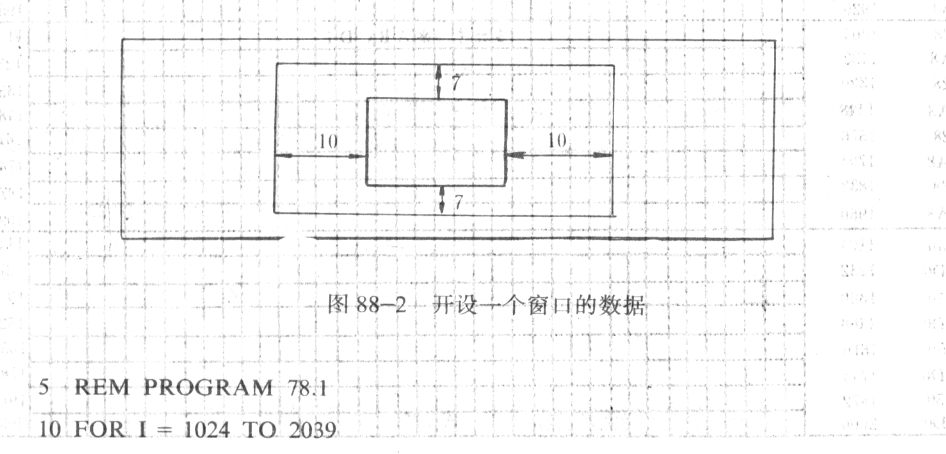


图 88-2　开设一个窗口的数据

```
5  REM PROGRAM 78.1
10 FOR I = 1024 TO 2039
20 POKE I,170
30 NEXT I
40 GET A$
50 POKE 32,10: POKE 33,20: POKE 34,7: POKE 35,17
60 HOME
70 VTAB 9: HTAB 10: PRINT "$"
```

89. 如何使用 POKE 语句在文本屏幕上写字符

中华学习机显示字符的过程是这样的：首先将字符的代码数据存入相应的内存单元中，然后计算机对内存单元中的数据进行译码并将其代码对应的字符显示在屏幕的相应位置上。可见，文本存贮区域中，存贮单元中的地址决定了在屏幕的什么位置显示字符，存贮单元中的数据定了屏幕显示什么字符。如果使用 POKE 语句往文本存贮区域中某个内单元中送一数据，就可在屏幕的某一位置显示一个字符。如果要想在屏幕的确定位置上显示某一确定的字符，就必须知道文本内存区域内存单元地址与屏幕上的字符位置的对应关系，以及字符与内存单元中的数值的对应关系。

关于字符与内存单元中数值的对应关系实际为字符与显示 ASCⅡ码的对应关系，可参看问题 24。

表 89–1 关于内存单元地址与文本屏幕上的字符位置的对应关系表

		列	1				5					10					15					20					25					30					35					40	
行	行地址码	列地址码	0	1				5					10					15					20					25					30					35				39	
1	$400	1024																																									1063
2	$480	1152																																									1191
3	$500	1280																																									1319
4	$580	1408																																									1447
5	$600	1536																																									1575
6	$680	1644																																									1703
7	$700	1792																																									1831
8	$780	1920																																									1959
9	$428	1064																$	=	!	%	*	A	B	C	D	E																1103
10	$4A8	1192																																									1231
11	$528	1320																																									1359
12	$5A8	1448																																									1487
13	$628	1576																																									1615
14	$6A8	1704																																									1743
15	$728	1832																																									1871
16	$7A8	1960																																									1999
17	$450	1104																																									1143
18	$4D0	1232																																									1271
19	$550	1360																																									1399
20	$5D0	1488																																									1527
21	$650	1616																																									1655
22	$6D0	1744																																									1783
23	$750	1872																																									1911
24	$7D0	2000																																									2039

关于内单元地址与文本屏幕上的字符位置的对应关系如表 89-1 所示。例如，与文本屏幕第四行第二列字符位对应的内存单元地址为 $ 581 (1409)。

如果要在文本屏幕第九行第十六列至第二十五列依次显示(以正常显示方式) $ #! % * ABCDE 十个字符，可运行下面的程序。

```
5    REM PROGRAM 89.1
10   FOR I= 1079 TO 1088
20   READ N: POKE I,N
30   NEXT I
40   DATA 164,163,161,165,170,129,130,131,132,133
```

由表 89-1 可以看出，内存地址号码与屏幕字符的对应关系并不是简单地顺序排列，那么它们是按什么规律排列的呢?

可以认为，将地址 1024～2047 (共 1024 内存地址) 从左至右、从上至下地顺序排列在图 89-1 所示的 A、B、C、D 四个区域中。然后，将 A、B、C 三个部分纵向排列即得到表 89-1 所示的文本地址分配图。D 区域内 64 个内存单元地址留给外设使用。

对于文本第二页内存，地址分配也存在上述规律，只是起始地址为 2048。

	40 个	40 个	40 个	8 个
8 行	1024........1063	1064........1103	1104........1143	1144........1151
	1152........1191	1192........1231	1232........1271	1272........1279
	·	·	·	·
	·	·	·	·
	·	·	·	·
	1920........1959	1960........1999	2000........2039	2040........2047
	A	B	C	D

图 89-1　文本内存单元分配的规律

注:文本地址号码排列与通常使用的 24 行 40 列坐标差别很大，不利于编写程序。为此，可建立一个子程序来将坐标 (X, Y) 转换成其对应的地址号码"P"。子程序如下:

```
5   REM PROGRAM 89.2
200  Y1 = Y - 1: M = INT (Y1 / 8)
210  P = 1024 + X - 1 + 40 * M + (Y1 - 8 * M) * 128
220  RETURN
```

90. 如何使用 POKE 语句绘低分辨率图形

中华学习机绘低分辨率图形的过程是这样的: 首先将色块的代码数据存入相应的内存单元中，然后计算机对内存单元中的数据进行译码并将其代码对应的色块显示在屏幕的相应位置上。因此，也需解决内存单元地址与屏幕色块位置的对应关系 (即内存单元地址分配特点)，以及内存单元中代码数据与色块的对应关系。

由于低分辨率第一页图形信息与文本第一页字符信息都存贮在同一内存区域，第二页低分辨率图形与第二页文本字符也同存贮在同一个内存单元中，所以低分辨率图形内存单元地址文本特点与文本内存单元地址分配特点一样 (参看问题"如何使用 POKE 语句在文

本屏幕上写字符”)，不同的是文本情况下显示一个字符的地方，在低分辨率图形显示情况下显示上下两个色块。

内存单元中的代码数据与色块颜色的对应关系是：内存单元低四位的数值决定下面小色块的颜色，高四位的数值决定下面小色块的颜色，如图90–1所示。数值与颜色的对应关系可参看表90–1。

表 90–1　代码与颜色的对应关系

代码	颜色	代码	颜色
$ 0(0)	黑	$ 8(8)	棕
$ 1(1)	紫红	$ 9(9)	橙
$ 2(2)	深蓝	$ A(10)	灰
$ 3(3)	紫	$ B(11)	粉红
$ 4(4)	墨绿	$ C(12)	浅绿
$ 5(5)	灰	$ D(13)	黄
$ 6(6)	青蓝	$ E(14)	碧绿
$ 7(7)	浅蓝	$ F(15)	白

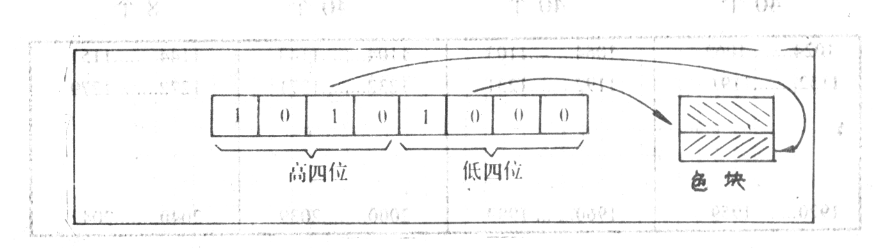


图 90–1　内存单元中数据与色块颜色的关系

举例如下：

在屏幕中间绘一朵红色小花（见图 90–2）。程序如下：

```
5    REM PROGRAM 90.1
10   GR
20   POKE 1338,17: POKE 1465,17
30   POKE 1467,17: POKE 1594,17
40   END
```

1465　1338　1467　1594　1466(第12行第19列)

图 90–2　绘一朵小红花

关于文本情况下坐标（X，Y）与地址P的转换子程序，在低分辨率绘图时也适用。

91. 如何用POKE语句绘高分辨率图形

中华学习机在高分辨率图形显示方式时，屏幕可以显示192行、每行280个色点，颜色可以是黑、白、紫、绿、红、蓝六种颜色中的一种。高分辨率绘图的实质也是向相应的内存单元送入数据，使屏幕上对应的位置显示出色点。

高分辨率图形的每个色点对应着存贮单元中的一位。每个存贮单元的八位中，第7位用于显示，各对应屏幕上一个色点，第八位用来确定颜色。7个色点在屏幕上是连续排列的，最左边的色点对应着存贮单元的最低位，左起第二个色点对应存贮单元的第二位……，如图91–1所示。一行280个色点对应着40个存贮单元，全屏幕192行共53760个色点，对应着7680个存贮单元。

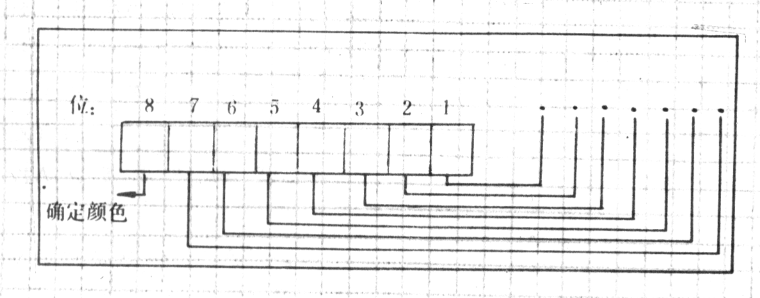


图91–1 色点与内存单元各位的对应关系

要用POKE语句往内存单元中送数据，以实现高分辨率绘图，也需要解决内存单元地址与屏幕色点的对应关系（即高分辨率图形内存单元地址分配特点）及内存单元中的数据与色点颜色的对应关系。

高分辨率图形内存单元地址分配特点与文本内存单元地址分配特点很相似，也分成A、B、C三部分，图91–2所示。不同的是图91–2中每个方块不是对应一个内存单元地址，而是包含有八个内存单元地址。图中列地址码和行地址码的和表示方块中最上方的内存单元地址。把方块中第一个内存单元的地址加上（n−1）*1024就可以得到方块中第n个内存单元地址了。运行下面的程序可以观察到高分辨率图形内存单元地址的分配特点。

```
5   REM  PROGRAM 91.1
10  HGR: POKE - 16302,0
20  FOR I = 0 TO 9207
30  POKE 8192 + I,  255
40  FOR J = 1 TO 100: NEXT J
50  NEXT I
60  END
```

高分辨率图形第二页的内存地址分配特点与第一页的情况一样，不同的是：第一页内存区域首地址为8192，而第二页内存区域首地址为16384。

色点的颜色与内存单元中的数据有什么对应关系呢？在黑白监视上，如果与色点对应的存贮单元相应位的值为 1，则这个点为白色；如果与色点对应的存贮单元相应位的值为 0，则这个点为黑色。在彩色监视器上，如果内存单元位的值为 0，则相应的点为黑色；如果内存单元位的值为 1，则相应的点为彩色。到底是什么颜色，还需看这个点在屏幕上属于哪一列（0～279 列），以及该内存单元最高位的值是 1 还是 0 来确定。如果最高位的值是 0，而且色点在第 0 列或其它偶数列，则颜色为紫色；如果色点在奇数列，则颜色为绿色。如果最高位的值是 1，色点在第 0 列或其它偶数列，则颜色为蓝色；色点在奇数列，则颜色为红色。如果两个色点紧挨着，哪怕两个色点所对应的位分属不同的内存单元，则它们均白色。可见，在高分辨率图形显示方式下，可以有六种不同的颜色。

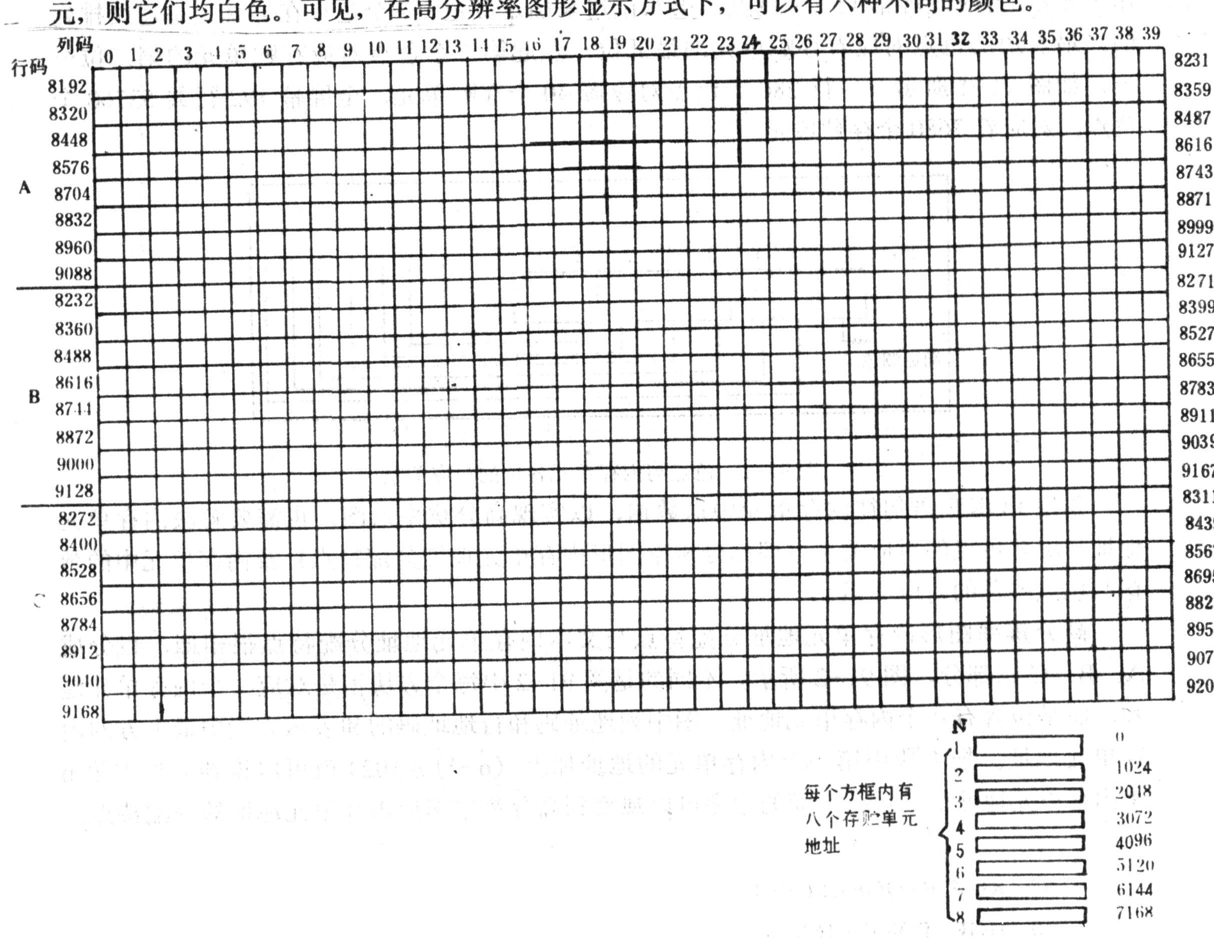


图 91–2　第一页分辨率图形存贮区域的内存地址与屏幕色点位置的对应关系

下面举两个例子。

例 1. 在屏幕中间显示由 7 个点组成的斜线。

利用纵向的七个内存单元，第一个内存单元地址为 8252，第 n 个内存单元地址为 8252+（n−1）x 1024。各内存单元中的数值按 91−3 所示的方法来确定：

n	存贮单元地址	十进制	二进制	图形
1	8252	1	00000001	□■■■■■■
2	9276	2	00000010	■□■■■■■
3	10300	4	00000100	■■□■■■■
4	11324	8	00001000	■■■□■■■
5	12348	16	00010000	■■■■□■■
6	13372	32	00100000	■■■■■□■
7	14396	64	01000000	■■■■■■□

图 91–3　斜线图案及相应内存单元中的数据

```
5  REM PROGRAM 91.2
10 HGR: POKE－16302, 0
20 FOR I= 8252 TO 14396 STEP 1024
30 READ N: POKE I, N
40 NEXT I
50 DATA 1, 2, 4, 8, 16, 32, 64
```

例 2.在屏幕中间显示一个“A”字。

根据图 91–4，可得到下面的程序。

N	存贮单元地址	十进制	二进制	图形
1	8252	8	00001000	■■■□■■■
2	9276	20	00010100	■■□■□■■
3	10300	34	00100010	■□■■■□■
4	11324	34	00100010	■□■■■□■
5	12348	62	00111110	■□□□□□■
6	13372	34	00100010	■□■■■□■
7	14396	34	00100010	■□■■■□■

图 91–4　字符"A"图案及相应内存单元中的数据

```
5  REM PROGRAM 91.3
10 HGR: POKE－16302,0
20 FOR I= 8252 TO 14396 STEP 1024
30 READ N: POKE I,N
40 NEXT I
50 DATA 8,20,34,34,62,34,34
```

为了能方便地在高分辨率显示屏幕的某行（对应文本显示方式下的1～24行）某列(对应文本显示方式下的1～40列）的方块处显示一个字符，可以象文本显示方式那样建立一个子程序。例如，在屏幕第10行、第20列显示一个“A”字，其程序如下：

```
5    REM PROGRAM 91.4
10   HGR：POKE－16302,0
20   X = 20: Y = 10
30   GOSUB 200
40   FOR I = P TO P +1024 * 6 STEP 1024
50   READ N, POKE I,N
60   NEXT I
70   END
100  DATA 8,20,34,34,62,34,34
200  Y1 = Y－1: M = INT (Y1 / 8)
210  P = 1024＋X－1＋40 * M +(Y1－8 * M) * 128
220  RETURN
```

92. 如何使中华学习机奏乐曲

中华学习机中设置了一个内存单元（地址为＄C030），每访问一次这个内存单元（读出该内存单元中的数据），就会使喇叭响一声。但如果利用BASIC语言编写发音程序，会因运行时间长而使发音音调很低。为获得各种音调的音乐声响应用机器语言编写发音程序。

机器语言发音子程序如下：

```
* 300L
0300-  98          TYA
0301-  00          BRK
0302-  AD  30  C0  LDA   $C030
0305-  88          DEY
0306-  D0  05      BNE   $030D
0308-  CE  01  03  DEC   $0301
030B-  F0  09      BEQ   $0316
030D-  CA          DEX
030E-  D0  F5      BNE   $0305
0310-  AE  00  03  LDX   $0300
0313-  4C  02  03  JMP   $0302
0316-  60          RTS
```

使用该机器语言发音子程序之前，应向＄300(768)内存单元中存放音调数据，向＄301(769)内存单元中存放节拍数据。然后，再用CALL 770调用该机器语言子程序使喇叭发出相应节拍与音调的声音。音调与节拍数据参看表92–1、表92–2。

表 92–1　音调数据

调＼音符	1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7
C					255	228	204	192	172	152	144	128	114	102	96	86	76	72	64	57	51
b D					242	216	192	182	162	144	136	121	108	96	91	81	72	68	60.5	54	48
D				255	228	204	182	172	152	136	128	114	102	91	86	76	68	64	57	51	45.5
bE			255	242	216	132	172	162	144	128	121	108	96	86	81	72	64	60.5	54	48	43
E			242	228	204	182	162	152	136	121	114	102	91	81	76	68	60.5	57	51	45.5	40.5
F		255	228	216	192	172	152	144	128	114	108	96	86	76	72	64	57	54	48	43	38
bG		242	216	204	182	162	144	136	121	108	102	91	81	72	68	60.5	54	51	45.5	40.5	36
G	255	228	204	192	172	152	136	128	114	102	96	86	76	68	64	57	51	48	43	38	34
bA	242	216	192	182	162	144	128	121	108	96	91	81	72	64	60.5	54	48	45.5	40.5	36	32

表 92–2　节拍数据

节　拍	十六分音符		八分音符		四分音符	
	$\underline{\underline{X}}$	$\underline{\underline{X}}\cdot$	$\underline{X}$	$\underline{X}\cdot$	X	X.
数　据	30	50	70	110	160	225

休止符数据为 1。

例如，编写使中华学习机奏下面乐曲的程序。

c 调	$\underline{1}$ ·	2	$\underline{1}$ |	$\underline{3}$	4	$\underline{3}$ |	5	0 |	i	0 |	5	3 |
节拍数据:	70	160	70	70	160	70	160	160	160	160	160	160
音调数据:	192	172	192	152	144	152	128	1	96	1	128	152

程序如下:

```
5    REM PROGRAM 92.1
10   HOME
20   FOR I = 770 TO 790
30   READ J
```

```
40   POKE I, J
50   NEXT I
60   DATA 173, 48, 192, 136, 208, 5, 206, 1, 3, 240, 9, 202, 208, 245, 174, 0,
     3, 76, 2, 3, 96
70   DIM P (12), D (12)
75   FOR I = 1 TO 12
80   READ P (I), D (I)
90   NEXT I
100  FOR I = 1 TO 12
110  POKE 768, P(I): POKE 769, D(I)
120  CALL 770
130  NEXT I
140  DATA 192, 70, 172, 160, 192, 70, 152, 70, 144, 160, 152, 70, 128, 160, 1, 160,
     96, 160, 1, 160, 128, 160, 152, 160
```

中华学习机 CEC-BASIC 语言还增加了一条用于奏乐曲。语句格式是:

MUSIC X, Y

其中 X 为音调数据(参看表 94-1), 取值范围为 0—255 之间; Y 为节拍数据(参看表 94-2), 取值范围也为 0—255 之间。

下面仍以上面的乐曲为例, 用 MUSIC 语句编写一个奏乐的程序。

```
10  READ X, Y
20  IF X=-1 THEN END
30  MUSIC X, Y : GOTO 10
40  DATA 192, 70, 172, 160, 192, 70, 152, 70, 144, 160, 152, 70, 128, 160, 1, 160, 96, 160,
    1, 160, 128, 160, 152, 160, -1, -1.
```

要使乐曲循环演奏, 可将 20 语句改为:

```
20  IF X=-1 THEN RESTORE: GOTO 10
```

93. 如何使中华学习机奏出长节拍的乐曲

由问题"如何使中华学习机奏乐曲"知道, 如何使中华学习机奏乐曲。但由于 $ 301 内存单元存放的最大数值只能是 $ FF (255), 所以发音节拍长度受到限制。为了使中华学习机发出音拍长的乐曲, 可用 $ 0301、 $ 0302 (769、770) 两个内存单元来存放节拍数据, 而音调数据可仍用 $ 300 (768) 内存单元来存放。为此, 机器语言发音子程序应作改动, 并将它存于 $ 0303 (771) 开始的内存单元中。

机器语言子程序如下:

```
* 303.32A
0303-  A9  04  20  A8  FC
0308-  AD  30  C0  88  D0  13  38  AD
0310-  01  03  E9  01  8D  01  03  AD
0318-  02  03  E9  00  8D  02  03  90
0320-  09  CA  D0  E7  AE  00  03  4C
0328-  03  03  60
```

节拍数据如表 93－1 所示。

表 93－1

节拍数据	十六分音符		八分音符		四分音符		二分音符		全音符
	X (=)	X. (=)	X (－)	X. (－)	X	X.	X	X－－	X－－－
	30	50	70	110	160	260	340	540	820

例如，编写使中华学习机奏下面乐曲的程序。

乐曲

bE 调　5 1　3 | 2 4　3 | 2 2　2 1 2 | 5 －－－ |

　　　6 7　1 | 1 2　3 | 5 4　3 1 | 2 －－－ |

程序如下：

```
5   REM  PROGRAM 93.1
10  FOR  I =  771  TO  810
20  READ  N: POKE  I,N
30  NEXT  I
40  DATA  169,4,32,168,252,173,48,192,136,208,19,56,173,1,3,233,1,141,1,3,173,2,3,233,0,141,
     2,3,144,9,202,208,231,174,0,3,76,3,3,96
50  DIM  P(23),D(23)
60  FOR  I = 1  TO  23
70  READ  P(I),D(I)
80  NEXT  I
90  FOR  I = 1  TO  23
100  POKE 768,P(I): POKE 769,D(I) - INT  (D(I) / 256) *  256: POKE 770,INT
  (D(I) / 256)
110  CALL  771
120  NEXT  I
```

```
200  DATA 216,70,162,70,128,160,144,70,121,70,128,160,144,70,144,70,144,70,162,30,144,
     30,216,340,132,70,172,70,162,160
210  DATA 162,70,144,70 128,160,108,70,121,70,128,70,162,70,144,340
```

94. 如何在屏幕上产生一串字符移动的效果

在屏幕同一行上下不断快速显示有一定规律的一个个字符串，利用人眼视觉暂留特点可产生字符移动的效果。例如:

第一次在第一行显示:　ABCDEFGHIJK

第二次在第一行显示:　BCDEFGHIJKA

第三次在第一行显示:　CDEFGHIJKAB

则在屏幕第一行可以产生 ABCDEFGHIJK 各字符由右向左移动的效果。

下面这个程序运行后，可在屏幕上方产生从右向左移动的一串字符，在下方产生从左向右移动的一串字符。

```
5   REM PROGRAM 94.1
10  HOME: N$ = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789":M$ = N$
30  N$ = MID$ (N$,2)+LEFT$ (N$,1)
40  M$ = RIGHT$ (M$,1)+MID$ (M$,1,35)
50  VTAB 4: HTAB 2: PRINT N$
60  VTAB 20: HTAB 2: PRINT M$
70  FOR J = 1 TO 100: NEXT J
80  GOTO 30
```

95. 如何编写产生活动图形的程序

一般活动的图形有两类,一类是图形简单地交替变化,例如眼球能来回移动的脸，嘴能一张一闭的鱼等；另一类是图形连续地移动，例如能行走的小人，能飞的小鸟等。实现图形简单地交替变化,可以在两个不同的绘图页上分别绘制两个状态不同的图形,然后利用屏幕显示软开关进行交替显示即可。例如能显示四只嘴一张一闭的小鱼的程序如下:

```
5   REM PROGRAM 95.1
10  HGR2 :HCOLOR= 3
20  FOR X = 10 TO 150 STEP 140
30  FOR Y = 48 TO 143 STEP 95
40  READ W$,X1,Y1
50  IF W$ = "N" AND K = 0 THEN RESTORE :GOTO 90
60  IF W$ = "M" THEN RESTORE : GOTO 90
70  IF W$ = "D" THEN HPLOT TO X+X1, Y+Y1: GOTO 40
```

```
80  HPLOT X+ X1,Y+Y1:GOTO 40
90  NEXT Y,X
100  IF K=0 THEN K=1:POKE 230,32:CALL 62450:GOTO 20
110  S=1-S:POKE-16299-S,0
120  FOR I=1 TO 100:NEXT I:CALL-198:GOTO 110
130  DATA U,12,-6,D,80,-40,D,80,-10,D,100,-20,D,100,20,D,80,10,D,80,40,D,12,
6,D,20,0,D,12,-6,U,24,-7,U,25,-7,U,24,-6,U,25,-6,N,0,0,D,12,-6,D,20,0,D,12,
6,D,0,0,D,20,0,M,0,0
```

程序中，第 110 语句和 120 语句是用来实现页面转换的。当 S=0 时,执行 POKE-16299,0,是设定第二页显示；当 S=1 时,执行 POKE-16300,0,是设定第一页显示。S 不断在取 1 或取 2 上交替变化，因而实现两页画面的交替显示。

实现图形连续移动的方法,是利用人眼的视觉暂留特性,在屏幕上显示一个图形后，把它抹去，再在另一位置上绘另一图形，再抹去......，如此继续下去，就象卡通片一样产生活动图形的效果。抹图形可用原图形颜色的补色来重绘该图形或用黑色来重绘原图形。下面是一个能产生小色块来回水平移动的程序：

```
5   REM PROGRAM 95.2
10  GR:HOME
20  A=0:B=39:C=1
30  FOR X=A TO B STEP C
40  COLOR=9:PLOT X,20
50  COLOR=0:PLOT X,20
60  NEXT X
70  IF N=0 THEN A=39:B=0:C=-1:N=1:GOTO 30
80  N=0:GOTO 20
```

为了不使绘图与抹图过程在屏幕上显示出来，可采用不显示绘图的方法（见问题“如何实现不显示的高分辨率绘图”）。下面是能产生小方块水平来回移动且不显示绘图与抹图过程的程序：

```
5   REM PROGRAM 95.3
10  HGR:HGR2:L=32:A=5:B=265:C=2:E=4
20  FOR X=A TO B STEP C
30  S=1-S:POKE-16299-S,0
40  L=96-L:POKE 230,L
50  HCOLOR=0:X1=X-E:GOSUB 200
60  HCOLOR=3:X1=X:GOSUB 200
70  NEXT X
90  K=A:A=B:B=K:C=-C:E=-E
```

```
100  GOTO 20
200  FOR Y = 50 TO54
210  HPLOT X1, Y TO X1 + 4,Y
220  NEXT Y
230  RETURN
```

程序中，L＝32,S＝1 是使机器处于第二页绘图第一页显示状态,L＝64,S＝0 是使机器处于第一页绘图第二页显示状态。

96. 如何通过键盘控制图形移动

计算机的使用者可通过键盘输入信息控制图形移动。通过键盘输入信息实现用户与计算机的信息传递，可通过 INPUT、GET 语句来实现,但采用这两种语句会使程序停止运行，使受控图形以外的其它图形能停止移动。为了使受控图形以外的其它图形能照常移动，可采用读取键盘输入单元中数据的方式。

中华学习机中设有专用的键盘接收内存单元，任何从键盘输入的字符，其 ASCII 码首先存入这个内存单元中。该内存单元地址为＄C000(49152 或−16384)。对这个内存单元进行查询可获得用户通过键盘输入的信息（读出所按键的 ASCII 码）。这个内存单元用低七位存放所按键的 ASCII 码，第八位存放是否按了键的标志，该位数值为 1 时表示按过了键，该位为 0 时表示没按键。因此，按键＄C000，内存单元的值应大于或等于 128。

在每次由键盘输入字符信息以前，要对＄C000 内存单元进行清除，其方法是给一个清键内存单元(＄C010 或 49168.−16368)送入一个数。它的清键作用并不是把＄C000 内存单元各位全清为 0，而是将第八位由 1 清为 0，使＄C010 内存单元的值小于 128。

在程序中使用键盘内存单元的方法可通过下面这个程序看出。

```
5   REM PROGRAM 96.1
10  P=PEEK(-16384)
20  IF P<128 THEN 10
30  POKE-16368,0
40  M=P-128:A$=CHR$(M)
50  PRINT P;":";A$
60  GOTO 10
```

该程序运行后，屏幕可显示出所按键的字符与＄C000 内存单元中相应数值的关系。从中也可以看出计算机获取用户输入信息的方法。下面举个例子说明如何编写能通过键盘控制图形移动的程序

例题：在屏幕上方显示一行从左向右移动的字符串，在屏幕下方显示一行从右向左移动的字符串，中间有一字符“＊”受键盘控制可上、下、左、右移动。当用户按“I”键时“＊”向上移动，按“M”键时“＊”向下移动，按“J”键时“＊”向左移动，按“K”键时“＊”向右移动。

```
5  REM PROGRAM 96.2
10  HOME :X0 = 20: Y0 = 12: X=X0:Y = Y0:HTAB X0:VTAB Y0:PRINT"*"
20  FOR I = 51 TO 90: A$ = A$ +CHR$(I):NEXT I:B$ = A$
30  A$ = MID$ (A$,2)+LEFT$ (A$,1): B$ = RIGHT$ (B$,1)+MID$(B$,1,39)
40  VTAB 1:PRINT B$:VTAB 23:PRINT A$;
50  P = PEEK(-16384): IF P < 128 THEN FOR J = 1 TO 20: NEXT J: GOTO 30
60  W$ = CHR$ (P-128)
70  POKE -16368,0
80  IF W$ = "I" AND Y> 2 THEN Y = Y-1: GOTO 130
90  IF W$ ="J" AND X>1 THEN X=X-1: GOTO 130
100  IF W$ = 'M"AND Y<22 THEN Y = Y+1: GOTO 130
110  IF W$ ="K" AND X < 40 THEN X = X+1: GOTO 130
120  GOTO 30
130  HTAB X0: VTAB Y0: PRINT " "
140  X0 = X:Y0 = Y: HTAB X0: VTAB Y0: PRINT "*"
150  GOTO 30
```

97. 如何用游戏控制器控制图形移动

游戏控制器是一个装有几个按钮开关和几个 150K 欧姆电位器（用操作杆或旋钮调节）的黑盒子。使用者可通过它们向计算机输入信息。

计算机内设有三个内存单元，用来存贮按钮开关输入的信息。当未按下按钮时，其对应的内存单元的数值为零，当按下按钮时，它所对应的内存单元的数值为 255。因此，对有关的内存单元进行查询，可以知道用户是否按下按钮开关与按钮开关对应的内存单元地址为：

开关 1: $C060 49249 -16287
开关 2: $C062 49250 -16286
开关 3: $C063 49251 -16285

用下面的程序可以验证：

```
5   PROGROM 97.1
10  P=PEEK(-16287)
20  IF P>128 THEN PRINT"NO! ":GOTO 10
30  PRINT " P = " ; P : GOTO 10
```

计算机还设有四个内存单元，用来存贮与电位器阻值对应的数值，电位器阻值在 0～150Ω 之间变化时，其对应的内存单元的数值在 0～255 之间变化。这四个内存单元的地址分别为：

模拟量 0:　$C064　49252　−16284
模拟量 1:　$C065　49253　−16283
模拟量 2:　$C066　49254　−16282
模拟量 3:　$C067　49255　−16281

用下面的程序可以验证:

```
5  PROGROM 97.2
10  P=PEEK(−16284)
20  PRINT "P=";P
30  GOTO 10
```

中华学习机的 FPBASIC 语言为读出这四个内存单元中的数值提供了一个专用函数，即 PDL 函数，其格式为:

PDL (算术表达式)

其中算术表达式的值　(取整) 表示模拟量的号码。因为模拟量有四个，所以算术表达式的值应为 0.1.2 或 3 四个数中的一个。采用 PDL 函数的验证程序为:

```
5  PROGROM 97.3
10  P=PDL (0)
20  PRINT "P=";P
30  GOTO 10
```

使用按纽开关可以使屏幕图形产生一个固定的变化，例如投炸弹、开炮、挖坑等。电位器阻值变化，可使其内存单元产生较宽范围的数值变化，这些数值可作为图形位置的参数从而实现用操作杆 (改变电位器阻值) 来控制图形的移动。

98. 如何使用高分辨率图形表绘图

高分辨率图形表绘图有使用方便、图形可以放大和旋转等优点。这种绘图法的基本思想是将计算机绘图归纳为八个动作 (用八种绘图矢量表示)，用代码分别表示这八个动作，去指导计算机绘图。

一. 绘图动作及其代码

规定的绘图动作和相应的绘图矢量与代码如表 98−1 所示。

表 98−1　绘图动作的规定

绘图矢量	代码	绘图动作
↑	0 0 0	不绘点上移
→	0 0 1	不绘点右移
↓	0 1 0	不绘点下移

续表 98-1

←	0 1 1	不绘点左移
↑	1 0 0	不绘点上移
→	1 0 1	不绘点右移
↓	1 1 0	不绘点下移
←	1 1 1	绘点右移

任何高分辨率图形都是由一些色点组成的,绘一个图形的过程可归为绘图动作的连续过程.因此可用一些绘图矢量和代码表示出绘图的过程.例如绘一个小正方形,其绘图矢量及代码为(参看图 98-1 和表 98-2):

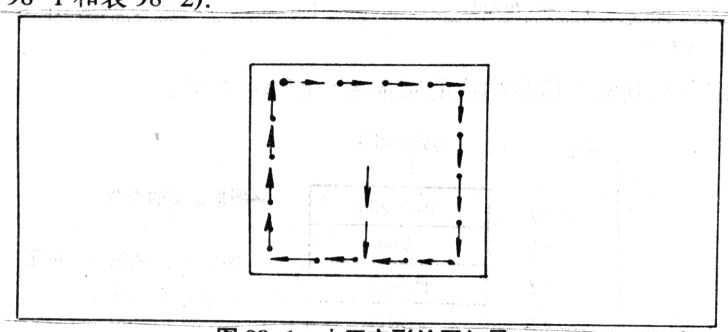


图 98-1　小正方形绘图矢量

表 98-2　小正方形绘图矢量与代码

绘图矢量	↓	↓	←	←	↑	↑	↑	↑	→	→	→	→	↓	↓	↓	↓	←	←
代码	010	010	111	111	100	100	100	100	101	101	101	101	110	110	110	110	111	111

二. 图形表的建立

将图形的绘图矢量代码按一定规律存入内存单元中就形成了图形定义，图形定义与有关索引项目按一定规律存放某一内存区域中就构成了图形表。

图形定义:

将绘图矢量代码依次存入内存单元中按下述规定进行:

(1)　八位的内存单元分为三段(见图 98-2)，第一、二段可以存放两个绘图矢量代码，第三段一般用来补两个零或填入一个不绘点的两位矢量代码(01 表示右移，10 表示下移，11 表示左移)。

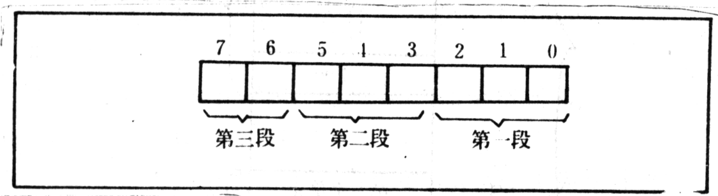


图 98-2　绘图矢量代码在内存单元中的存放情况

(2)　如果第三段内填入 00，则表示此段不起作用，如果第三、二段均为零时，表示这两段都不起作用。如果三段均为零，则表示该图形定义结束。

根据上述规定，上面小正方形的图形定义如表 98-3 所示:

表 98-3　小正方形图形定义

序号	内存单元中的绘图矢量代码	十六进制数	十进制数
1	00010010	$ 12	18
2	00111111	$ 3F	63
3	00100100	$ 24	36
4	00100100	$ 24	36
5	00101101	$ 2D	45
6	00101101	$ 2D	45
7	00110110	$ 36	54
8	00110110	$ 36	54
9	00111111	$ 3F	63
10	00000000	$ 00	0

2. 图形表的建立

若干图形定义与其索引信息构成的图形表如图 98-3 所示:

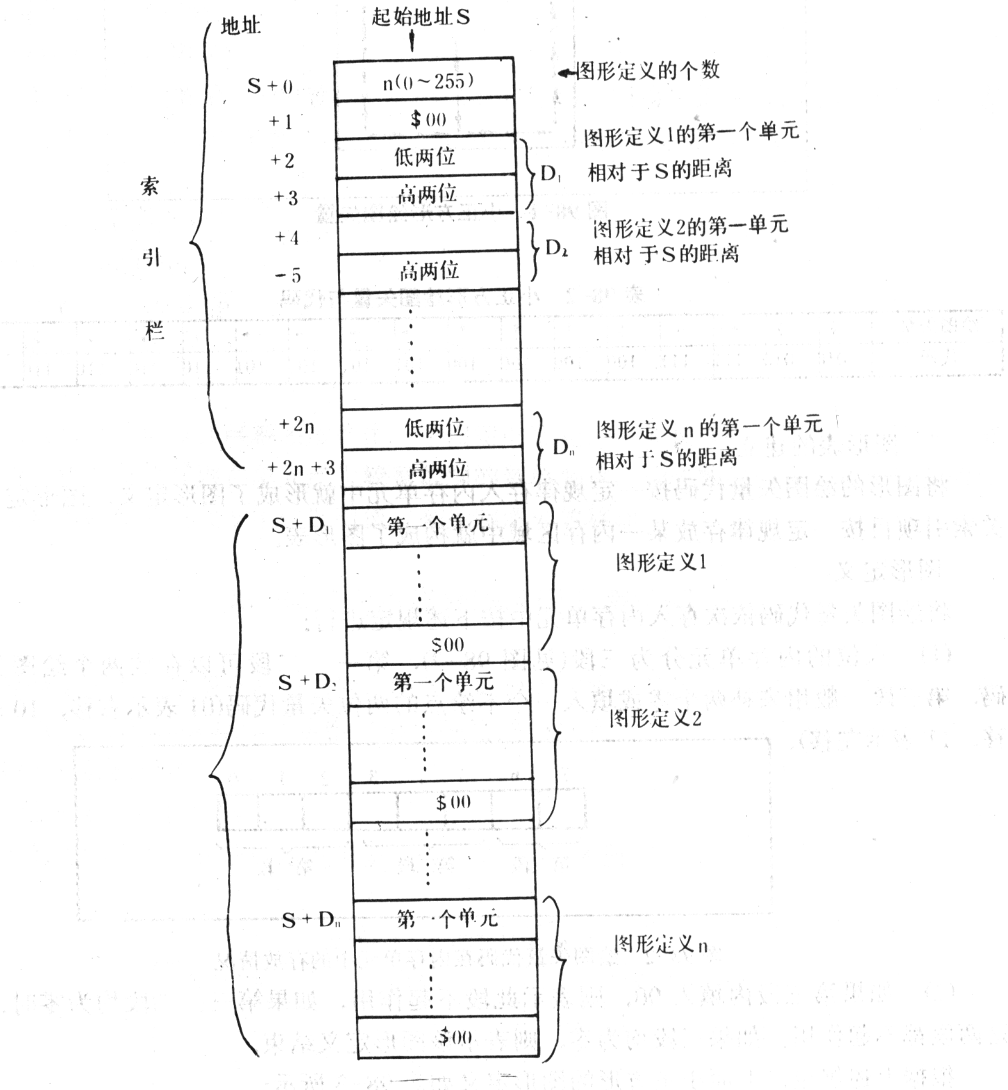


图 98-3　图形表

图形表中的图形定义栏可以和索引栏紧接着，也可以分开一段距离。下面小正方形的图形表如图 98–4 所示。

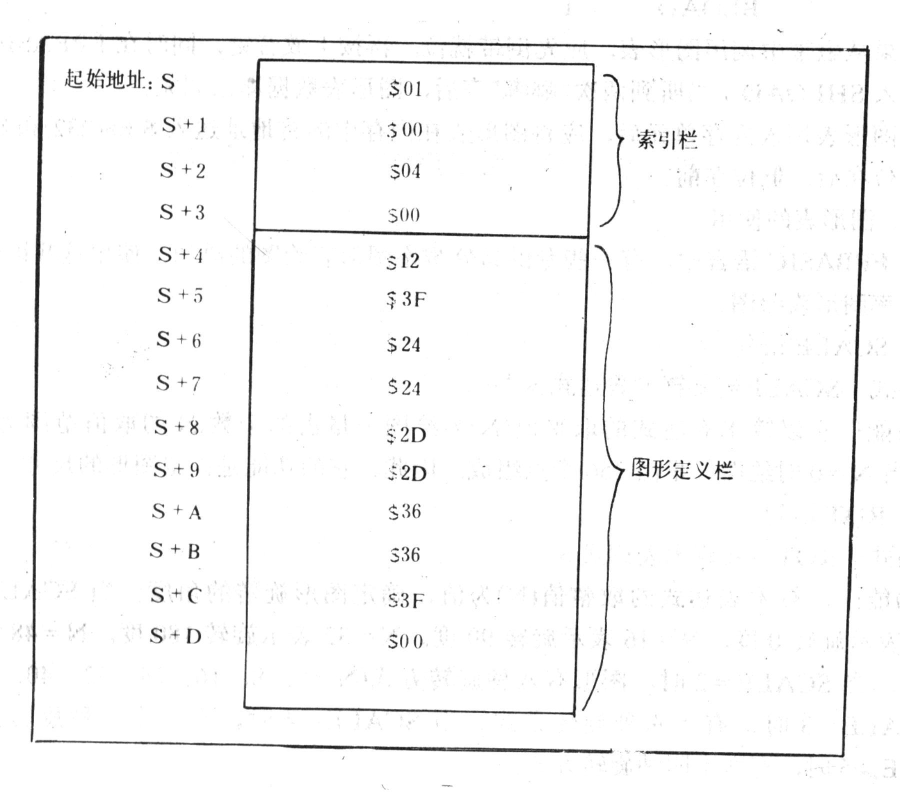


图 98–4　小正方形图形表

三. 图形表的存储与调用

1. 内存区域的选择

图形表所占内存区域的选择是以不被 BASIC 程序字符串和图形数据所破坏为依据的。因此，当图形表较小时，可存放在 $ 300 开始的内存单元中；当程序不太长，图形表不太大时，可存放在 $ 0C00– $ 1FFF 内存单元中；当图形表较大时，可存放在 $ 6000 开始的内存单元中。为了不使字符串数据破坏图形表，可用 HIMEM 语句重新疫定新的 BASIC 程序使用的上限内存单元地址.将图形表存贮在该内存单元的上方。

2. 存入磁盘或磁带中

(1)　存入磁盘：利用 DOS 命令 BSAVE 进行存贮,命令格式为：

BSAVE　　　f，Aa，Ll

其中，a 为图形表首地址,l 为图形表占用的内存单元个数，f 为文件名。

(2)　存入磁带：首先将图形表首末地址的差值（即图形表占用内存单元的个数减 1)，存入地址为 $ 00（存差值的低位）和 $ 01（存差值的高位）内存单元中。然后，在监控状态下键入：

＊0.1W <图形表起始地址>.<图形表尾地址>W

键完上述字符后暂不按回车键，装好磁带，倒带就位，置录音方式后再按回车键。

3. 图形表的调用

如果从磁盘中调用图形表可使用 BLOAD 命令，其格式为:

BLOAD　　f

如果从磁带中调用图形表，应先倒带就位，再按下放音键，同时在 FPBASIC 语言状态下键入 SHLOAD↓当听到两次"嘟嘟"声后，图形表数据即已调完。

将图形表调入内存单元后，应将图形表在内存中的首地址送入 $ E8(232)和 $ E9(233)中，高位在后，低位在前。

四. 图形表的使用

在 FPBASIC 语言中，有一些专供高分辨率图形表绘图的语句，使用这些语句可完成高分辨率图形表绘图。

1. SCALE 语句

格式: SCALE＝<算术表达式>

功能: 它以算术表达式的取整值(N)为绘图矢量占的个数,N 的取值范围为 0⩽N⩽255，当 N＝0 时绘图矢量由 256 个点组成。因此，它的功能是设定图形的尺寸。

2. ROT 语句

格式: ROT＝<算术表达式>

功能: 以算术表达式的取整值(N)为值，确定图形旋转的角度。当 SCALE＝1 时，N＝0 表示旋转 0 度，N＝16 表示旋转 90 度。N＝32 表示旋转 180 度，N＝48 表示旋转 270 度。当 SCALE＝2 时，图形有八种旋转方式(N＝0、8、16、24、32、40、48、56)。当 SCALE＝3 时，有十六种旋转方式。当 SCALE＝4 时，有三十二种旋转方式。当 SCALE⩾5 时，有六十四种旋转方式。

3. DRAW 语句

格式: DRAW (算术表达式 1) AT (算术表达式 2)，(算术表达式 3)

功能: 首先计算各表达式的值，并取整得到数值 N X Y(1⩽N⩽255,0⩽X⩽279,0⩽Y⩽191),确定了取第 N 个图形,图形起点为(X,Y)。当格式中没有 AT 以及其后各项时,图形起点是刚刚绘完图的结束点,若未绘图则图形起点定为(0,0)

4. XDRAW 语句

格式: XDRAW(算术表达式 1)AT(算术表达式 2),(算术表达式 3)

功能: 和 DRAW 的功能一样,只是图形颜色取原颜色的补色。颜色的互补关系为，

黑←白	蓝→绿	紫→橙
黑→白	蓝←绿	紫←橙

使用上述语句编写的高分辨率图形表绘图程序实例如下:

绘不断放大同时不断转动角度的正方形图案，其图形表(见图 98—4)存贮在以 $ 300 为起始的内存区域中。程序如下:

```
5  REM PROGRAM 98.1
10  PRINT CHR $ (4)"BLOAD HPLOT-DRAW"
20  POKE 232,0: POKE 233,3
30  HGR 2: HCOLOR = 3
```

```
40  FOR I =  1 TO 50
50  ROT = I: SCALE = I
60  DRAW 1 AT 139,79
70  NEXT I
80  END
```

99. 如何通过键盘操作建立图形定义表

运行下面的程序一，按提示操作，可通过按键 I（向上），K（向右），M（向下），J（向左），Y（绘点），N（不绘点），E（结束）来建立图形定义表。运行下面的程序二，按提示操作，可在屏幕上的方格区域（最大为 15 * 15 个小方格）内绘一个图形，并可以将此图形的图形表在内存某一区域内建立起来，而且还能将图形表存入磁盘或打印出来。

程序一如下：

```
5  REM PROGRAM 99.1
10  DIM SI (100),VI(100)
20  HOME: I = 0
30  PRINT "CREATE SHAPE VECTORS"
40  PRINT
50  V = I: GOSUB 270
60  IF P$ <  > "E" THEN SI(I) = M: I = I + 1: GOTO 50
70  PRINT
80  INPUT "VECTOR TO CHANGE(0 = END):";V
90  IF V > 0 THEN V = V - 1: GOSUB 270:SI(V) = M: GOTO 80
99  B = 0:Q = 0
100  FOR V = 0 TO I
110  IF B = 2 AND SI(V) > 0 AND SI(V) < 4 THEN 140
120  IF B < 2 AND (SI(V) > 0 OR SI(V) > 4) THEN 140
130  B = 0:Q = Q + 1
140  VI(Q) = VI(Q) + SI(V) * (8^B)
150  B = B + 1
160  IF B > 2 THEN B = 0:Q = Q + 1
170  NEXT V
180  PRINT "BYTE","VECTOR","DEC"
190  FOR V = 0 TO Q
200  H% = VI(V) / 16
210  L% = VI(V) - H% * 16
220  IF H% > = 10 THEN H% = H% + 7
```

```
230  IF L% > = 10 THEN L% = L% + 7
240  PRINT V, CHR $ (H% + 48); CHR $ (L% + 48),VI(V)
250  NEXT V
260  END
270  PRINT "VECTOR";V + 1;":";
280  INPUT "PLOT (Y = YES,N = NO,E = END)?";P $
290  IF P $ = "Y" THEN M = 4
300  IF P $ = "N" THEN M = 0
310  IF P $ = "E" THEN RETURN
320  INPUT "MOVE:I / M / J / K?";M $
330  IF M $ = "I" THEN M = M + 0: RETURN
340  IF M $ = "K" THEN M = M + 1: RETURN
350  IF M $ = "M" THEN M = M + 2: RETURN
360  IF M $ = "J" THEN M = M + 3: RETURN
370  GOTO 320
```

程序二如下：

```
1  REM PROGRAM 99.2 (SHAPE CONSTRUCTION)
2  REM ASSISTS IN CONSTRUCTING SHAPE TABLES
10  HCOLOR= 3: SCALE= 1: ROT= 0:TA = 24576
20  POKE 768,1: POKE 769,0: POKE 770,4: POKE 771,0
30  POKE 772,58: POKE 773,36: POKE 774,45: POKE 775,54: POKE 776,7: POKE 778,0
40  TEXT : HOME : PRINT "CHOOSE :": PRINT
50  PRINT "L—> LOAD A SHAPE TABLE FROM DISK": PRINT
60  PRINT "N—> START A NEW SHAPE TABLE": PRINT
70  PRINT "YOUR CHOICE"; CHR $ (95);: HTAB 12: INPUT A $
80  IF A $ = "N" THEN 180
90  IF A $ < > "L" THEN 40
100  PRINT : INPUT "NAME OF THE TABLE ";N $
110  PRINT CHR $ (4);"BLOAD ";N $ ;",A $ 6000"
120  N = ( PEEK (TA + 2) + 256 * PEEK (TA + 3) − 2) / 2
130  SN = PEEK (TA): PRINT : IF SN = N THEN PRINT "TABLE FULL": GOTO 2000
140  PRINT "THIS TABLE CAN HOLD ";N;"SHAPES": PRINT
150  PRINT " IT PRESENTLY HAS";SN;"SHAPE(S) IN IT": PRINT
160  FOR I = 1 TO 2500: NEXT I
170  GOTO 240
180  TEXT : HOME : INPUT "NUMBER OF SHAPES IN THIS TABLE:";N
190  POKE TA,0: POKE TA + 1,0
```

```
200  DA = 2 * N+2
210  POKE TA + 2,DA - 256 * INT (DA / 256)
220  POKE TA + 3, INT (DA / 256)
230  FOR I = TA + 4 TO TA + 2 * N + 3: POKE I,0: NEXTI
240  PRINT : PRINT "CHOOSE SIZE OF SHAPE DESIGN GRID"
250  INPUT "NUMBER OF COLMNS (1 - 15) ";C:C = 10 * C
260  INPUT "NUMBER OF ROWS (1 - 15) ";R:R = 150 - 10 * R
270  HGR
280  FOR I = R TO 150 STEP 10: HPLOT 0,I TO C,I: NEXT I
290  FOR I = 0 TO C STEP 10: HPLOT I,R TO I,150: NEXT I
300  IF A = 1 THEN 330
310  HOME: VTAB 21: PRINT "STARTING POSITION ? LOWAER LEFT IS (1,1)"
320  INPUT "COLUMN ";X1: INPUT "ROW ";Y1
330  X = 10 * X1 - 5: Y = 155 - 10 * Y1
340  ADDR = PEEK (TA + SN * 2 + 2) + 256 * PEEK (TA + SN * 2 + 3) + TA
350  POKE 232,0: POKE 233,3
360  XDRAW 1 AT X,Y:P = 0:V = - 1
370  HOME: VTAB 21: PRINT "TABLE CAPACITY:";N;"SHAPES:THIS IS # ";SN +
1    380  VTAB 22: PRINT TAB( 10);"DIRECTI ON (I / M / J / K)"
390  VTAB 23: PRINT "CHOOSE:PLOT (P)"
400  VTAB 24: PRINT TAB( 10);"QUIT THIS SHAPE (Q)";
410  VTAB 23: HTAB 35: GET V $ : PRINT
420  IF V $ = "Q" THEN POKE ADDR,P: POKE ADDR + 1,255: GOTO 520
430  XDRAW 1 AT X,Y
440  IF V $ = "P" THEN P = 4: FOR I = X - 3 TO X + 3: HPLOT I,Y - 3 TO I,Y +
3:  NEXT I: XDRAW 1 AT X,Y: GOTO 410
450  IF V $ = "I" THEN V = P:Y = Y - 10: IF Y < R THEN Y = Y + 10: PRINT CHR $
(7);:V = - 1
460  IF V $ = "K" THEN V = P + 1:X = X + 10: IF X > C THEN X = X - 10: PRINT
CHR $ (7);:V = V -1
470  IF V $ = "M" THEN V = P + 2:Y = Y + 10: IF Y > 150 THEN Y = Y - 10: PRINT
 CHR $ (7);:V = - 1
480  IF V $ = "J" THEN V = P + 3:X = X - 10: IF X < 0 THEN X = X + 10: PRINT
CHR $ (7);:V = - 1
490  IF V = - 1 THEN XDRAW 1 AT X,Y: GOTO 410
500  POKE ADDR,V:ADDR = ADDR + 1
510  XDRAW 1 AT X,Y:P = 0:V = - 1: GOTO 410
520  ADDR = PEEK (TA + SN * 2 + 2) + 256 * PEEK (TA + SN * 2 + 3) + TA:
BYTE = ADDR
```

```
530  C1 = PEEK (BYTE): IF C1 = 255 THEN POKE ADDR,0:ADDR = ADDR + 1:
     GOTO 660
540  C2 = PEEK (BYTE + 1): IF C2 = 255 THEN POKE ADDR,C1: POKE ADDR + 1,0:
ADDR = ADDR + 2: GOTO 660
550  C3 = PEEK (BYTE + 2): IF C3 = 255 THEN POKE ADDR,C1 + 8 * C2: POKE
ADDR + 1,0:ADDR = ADDR + 2: GOTO 660
560  CODE = C1 + 8 * C2 + 64 * C3
570  IF CODE = 0 THEN POKE ADDR,64: POKE ADDR + 1,24: ADDR = ADDR +
     1:BYTE= BYTE + 2: GOTO 650
580  IF CODE < 8 THEN POKE ADDR,CODE: GOTO 650
590  IF CODE < 64 THEN POKE ADDR,CODE:BYTE = BYTE + 1: GOTO 650
600  IF CODE < 256 THEN POKE ADDR, CODE:BYTE = BYTE + 2: GOTO 650
610  CODE = CODE - 64 * C3
620  IF CODE = 0 THEN POKE ADDR,64: POKE ADDR + 1,3:ADDR = ADDR +
     1:BYTE = BYTE + 1: GOTO 650
630  IF CODE < 8 THEN POKE ADDR,CODE: GOTO 650
640  POKE ADDR, CODE:BYTE = BYTE + 1
650  ADDR = ADDR + 1:BYTE = BYTE + 1: GOTO 53
660  POKE TA,SN + 1: POKE 233,96: DRAW SN + 1 AT 200,100
670  HOME : VTAB 22:PRINT "SAVE THIS AS SHAPE NUMBER";SN + 1;
"(Y / N)";: INPUT A $
680  IF A $ = "Y" THEN SN = SN + 1: IF SN < N THEN DA=ADDR-TA: POKE
TA+2 * SN+2,DA-256 * INT(DA / 256): POKE TA+2 * SN+3,INT(DA / 256)
690  I FA $ < > "N" AND A $ <  > "Y" THEN 670
700  POKE TA,SN
710  TEXT : HOME : VTAB 15: PRINT "CHOOSE:"
720  PRINT  TAB( 5);"(1) DRAW ANOTHER SHAPE"
730  PRINT  TAB( 5);"(2) CHANGE DOT MATRIX"
740  PRINT  TAB( 5);"(3) CHANGE STARTING POINT"
750  PRINT  TAB( 5);"(4) SAVE SHAPE TABLE TO DISK"
760  PRINT  TAB( 5);"(5) PRINTER SHAPE TABLE"
765  PRINT  TAB( 5);"(6) LEAVE THE PROGRAM"
770  INPUT A $ :A = VAL (A $ )
780  IF A < 1 OR A > 6 THEN 710
790  IF SN = N AND A < 4 THEN PRINT "TABLE FULL": FOR I = 1 TO 1000:
     NEXT I: GOTO  710
800  ON A GOTO 270,240,270,820,1000
810  TEXT : HOME : END
820  INPUT "NAME";N $
```

```
830  PRINT  CHR$ (4);"BSAVE ";N$",A24576,L";ADDR - 24576
840  GOTO 710
1000  D$ = CHR$ (4)
1010  H$ = "0123456789ABCDEF"
1020  PRINT D$;"PR#1"
1030  PRINT "ADDRESS","DEC","HEX"
1040  FOR JJ = 24576 T$O ADDR
1045  BASE = 16:II = 1
1050  NN = PEEK (JJ):NN = INT (NN)
1060  QQ = INT (NN / BASE)
1070  AA(II) = NN - BASE * QQ:II = II+1
1080  IF QQ > BASE THEN NN = QQ:GOTO 1060
1090  AA(II) = QQ
1100  PRINT JJ,NN,"$";
1110  FOR J = II TO 1 STEP - 1
1120  PRINT  MID$ (H$,AA(J) + 1,1);
1130  NEXT J
1140  PRINT
1150  NEXT  JJ
1160  PRINT D$;"PR#0"
1170  GOTO 710
2000  FOR I = 1 TO 2000: NEXT I
2010  A = 24576:K = 1:S = 1
2020  A1 = INT (A / 256):A2 = A - A1 * 256
2030  POKE 232,A2: POKE 233,A1
2040  HOME
2050  HGR
2060  ROT= 0: SCALE= 1:I = 1:K = 1
2070  HCOLOR = 3:X1 = 19 * I - 19:Y1 = 19 * K
2075  HPLOT X1,Y1 TO X1 + 19,Y1 TO X1 + 19,Y1 - 19 TO X1,Y1 - 19 TO X1,Y1
2079  HCOLOR= 3
2080  DRAW S AT 19 * I - 17,19 * K - 1
2090  I = I + 1:S = S + 1
2100  IF I = 15 THEN K = K + 1: I = 1
2110  IF S - 1 = 112 THEN VTAB 23: HTAB 12: PRINT "PRESS BREAK BAR": GET A
      $: GOTO 2050
2120  IF S - 1 = N THEN END
2130  GOTO 2070
```

100. 如何用 PRINT 语句在低分辨率图形显示方式下绘制低分辨率图形

由于文本显示与低分辨率图形显示共用一个内存区域,所以内存单元中某一确定的数值使计算机在文本显示情况下显示一个字符,而在低分辨率图形显示方式下显示上.下两个小色块,所以文本显示情况下的字符与低分辨率图形显示情况下色块的颜色存在一定的关系。这一关系可用表 100−1 来表示。

表 100−1　字符与色块颜色的代码

	字符显示方式				上面色块颜色的代码															
下面色块颜色代码	反转	闪烁	正常		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0	4	8	12	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	1	5	9	13	P	Q	R	S	T	U	V	W	X	Y	Z	[		]	ˆ	–
	2	6	10	14	!	″	#	$	%	&	′	(	)	*	+	,	–	.	/	
	3	7	11	15	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?

文本情况下，用 PRINT 语句写字符实质是将字符的显示 ASCII 码送入相应的内存单元中。因此，我们可以在低分辨图形显示方式下，用 PRINT 语句往相应内存单元中送数达到用 PRINT 语句绘低分辨率图形的目的。PRINT 语句打印的字符，应由需绘的低分辨率图形的颜色来决定,这可以通过查表 100−1 来完成。举例如下：在低分辨率图形显示方式下，以浅绿色为底色，粉红色写一"北"字。程序如下：

```
5   REM  PROGRAM 100.1
10   HOME : POKE – 16304,0: POKE – 16302,0
20   VTAB  10
30   PRINT  TAB (   10)″LLLLLLLLLLLLLLLLLL″
40   PRINT  TAB (   10)″LLLLLL;;LLL;;LLLLLL″
50   PRINT  TAB (   10)″LLLLLL;;LLL;;LLLLLL″
60   PRINT  TAB (   10)″L;;;;;;LLL;;;;;;L″
70   PRINT  TAB (   10)″LLLLLL;;LLL;;LLLLLL″
80   PRINT  TAB (   10)″LLLLLL;;LLL;;LLLLLL″
90   PRINT  TAB (   10)″L;;;;;;LLL;;;;;;L″
100   PRINT  TAB(   10)″LLLLLLLLLLLLLLLLLL″
110   GOTO 110
```

程序中，正常显示的字符 L 的显示 ASCII 码为 $ CC(204)，其 $ C 对应的颜色是绿色；正常显示的字符；的显示 ASCII 码为 $ BB(187)，其 $ B 对应的颜色是粉色。这

通过查表 100−1 即可得出。

101. 如何在文本情况下使用低分辨率绘图语句和 SCRN 函数

低分辨率绘图语句 PLOT，HLIN 和 VLIN 绘图的实质是往相应的内存单元的高四位或低四位送数据,而函数 SCRN 的实质是查询与坐标(X，Y)(指低分辨率图形显示的坐标)相应的内存单元低四位或高四位的数值。由于文本内存单元地址与低分辨率图形内存单元地址是一样的，而且有相同的分配特点，因此，可以在文本情况下使用上述各条低分辨率绘图语句和函数。

在文本显示情况下使用 PLOT.HLIN.VLIN 语句和 SCRN 函数时,除了应注意字符与颜色的对应关系外,还应注意文本显示情况下坐标位置与低分辨率图形显示情况下坐标位置的对应关系。下面举几个例子说明在文本显示情况下如何用低分辨率绘图语句与 SCRN 函数。

例 1. 在文本显示状态下用 HLIN 语句绘下述的字符图案。

```
AAA
AAAAA
AAAAAAA
AAAAAAAAA
AAAAAAAAAAA
AAAAAAAAAAAAA
AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

程序如下：

```
5   REM PROGRAM 101.1
10  HOME
20  FOR Y = 2 TO 39
30  IF Y / 2 = INT (Y / 2 )THEN COLOR = 1: HLIN 0, Y AT Y:GOTO 50
```

```
40  COLOR = 12: HLIN 0, Y - 1 AT Y
50  NEXT Y
```

例 2. 读出文本屏幕任一位置的字符。

程序如下:

```
5  REM PROGRAM 101.2
10  INPUT X,Y
20  N$ = CHR$(SCRN(X,(Y-1) * 2) + SCRN(X,(Y-1) * 2+1) * 16)
30  PRINT N
```

例 3. 将文本屏幕上的内容打印出来。

```
3  REM PROGRAM 101.3
5  PR#1
10  FOR Y = 0 TO 47 STEP 2
20  FOR X = 0 TO 39
30  N = SCRN (X, Y) + SCRN(X, Y + 1) * 16
40  PRINT CHR$(N);
50  NEXT X: PRINT
60  NEXT Y
70  PR# 0
```

102. 如何用 FPBASIC 程序迁移内存单元的信息

我们常需要将某一内存区域中各内存单元的数据依次迁移至另一内存区域的各内存单元中。这种迁移内存单元信息的工作，通常可采用以下三种方法。

一. 利用 POKE,PEEK 语句

例如要将高分辨率第二页的图形信息($4000-$5FFF 内存单元中的数据)迁移到高分辨率第一页内存区域($2000-$3FFF)中可执行以下程序:

```
5  REM PROGRAM 102.1
10  FOR I = 8192 TO 16383
20  POKE I, PEEK(I+8192)
30  NEXT I
```

这种方法简洁,清楚.但运行速度很慢,只适用于小量的内存信息迁移。

二. 利用 MOVE 迁移子程序

可以利用监控程序中的 MOVE 迁移子程序使迁移速度大大加快。具体方法如下:

(1) 首先在 $300-$304 内存区域中建立一个机器语言子程序，它完成给寄存器 Y 赋初值和转至 MOVE 子程序的工作。程序如下:

```
*300-A0 00        LDY    #$00
```

```
*302-4C 2C FE JMP        $FE2C
```

用 FPBASIC 程序输入该机器语言程序时，程序为：

```
5  REM PROGRAM 102.2
10  POKE 768,160: POKE 769,0
20  POKE 770, 76: POKE 771,44
30  POKE 772,254
```

(2)　赋迁移指针数据

将要迁移的信息所在的存贮区域(用 SOURCE 表示)的首,尾地址,以及接受信息的存贮区域(用 DESTIN AION 表示)的首地址存入相应的内存单元中,参看表 102-1。

(3)　执行 CALL768,使计算机执行 MOVE 迁移子程序。

使用上述方法时应注意:DESTINAION 的起始地址不应在 SOURCE 中，否则 SOURCE 内的部分信息会在未迁移前被更改。

下面的程序可完成移高分辨率第一页内存区域的工作。

表 102-1

地　址	数　据
$3C　(60)	SOURCE 起始地址低位
$3D　(61)	SOURCE 起始地址高位
$3E　(62)	SOURCE 终止地址低位
$3F　(63)	SOURCE 终止地址高位
$42　(66)	DESTINAION 起始地址低位
$43　(67)	DESTINAION 起始地址高位

```
5  REM PROGRAM 102.3
10  POKE 768,160: POKE 769,0
20  POKE 770,76:POKE771,44
30  POKE 772,254
40  POKE 60,0: POKE 61, 64
50  POKE 62,255: POKE 63,95
60  POKE 66,0: POKE 67,32
70  CALL 768
```

三. 利用监控命令的方法

在监控方式下有相应的迁移信息命令来完成内存信息迁移工作，例如将高分辨率第二页图形信息迁移至高分辨率第一页内存区域中,可采用以下方法:

```
*2000<4000.SFFFM
```

在 BASIC 程序中可执行上述命令(参看问题 54)。

例如,完成上述高分辨率图形信息迁移的 BASIC 程序为:

```
5  REM PROGRAM 102.4
10  Y$ = "2000 < 4000.SFFFM N D 823 G"
20  FOR I = 1 TO LEN ( Y$ )
30  POKE 511 + I, ASC(MID$(Y$,I,1))+ 128
40  NEXT I: POKE 72, 0
50  CALL - 144
```

103. 如何在文本第二页写字符

通常不能直接在文本第二页写字符,这是因为中华学习机文本第二页所处的内存区域 $ 800－$ BFF,正好与 BASIC 程序区相重叠。如果要在文本第二页写字符，会使 BASIC 程序遭到破坏。为了能在文本第二页写字符，可将程序区移至其它地方，移动的方法是改变程序首地址指针和其它有关数据。例如将程序区改为 $ 6001 为起始地址的内存区域中，可依次键入以下命令:

```
POKE   103,1: POKE 103, 96
POKE   24576,0
NEW
```

其中 103($ 67)存程序区首址低位,104(68)存程序区首址高位,24567($ 6000)内存单元需置零。键入 NEW 命令的作用是让中华学习机按 $ 67、$ 68 指示的程序区改变其它有关指针数据 。

新程序区设定后就可以在文本第二页上写字符了，写字符的方法有以下两种:

一. 内存信息迁移法

这种方法是先在文本第一页写好字符，然后再将它迁移到文本第二页，也就是将 $ 400－$ 3FF 内存单元中的数据依次迁移到 $ 800－$ BFF 内存区域中。具体迁移的方法可参看问题″如何迁移内存单元的信息″。

二. 直接写字符法

在设定进行第二页写字符后，可直接在第二页写字符的方法是将内存单元 $ 29(41)中的数据改为 8－11 之间的一个数即可($ 29 内存单元内的数值正常时为 1－7 之间的数,设定的是在文本第一页写字符)。在进行第二页写字符时需注意每执行一个回车键后都会使 $ 29 内存单元中的数复原,也就是恢复设定在文本第一页写字符。因此，应注意在执行回车后将 $ 29 内存单元置 8－11 之间的一个数。

在文本第二页写字符的实例如下:

先将程序 1 以名字 TEXT－2 存入磁盘，再运行程序 2,即可在文本第一页绘″七一″两字，在文本第二页绘“十一”两字，然后交替显示。

程序 1 如下:

```
5   REM PROGRAM 103.1
10  HOME : VTAB 7: HTAB 17:INPUT "A $ =";A $ : HTAB 17: INPUT "B $ =";B $
20  FOR I = 2048 TO 3071: POKE I, 160:NEXT I: HOME
30  VTAB 5: POKE - 16300,0:K = 0:N $ = A $ : GOSUB 200
40  VTAB 5: POKE - 16299, 0:K = 4:N $ = B $ : GOSUB 200
50  S = 1 - S: POKE - 16299 -S,0: CALL - 198
60  FOR J = 1 TO 1000: NEXT J : GOTO 50
```

```
200  PRINT : POKE 41, PEEK (41) + K
210  READ X
220  IF  X < 0 THEN 200
230  IF X > 40  THEN :RETURN
240  HTAB  X
250  READ  Y
260  FOR I = X TO Y:PRINT N $ ;: NEXT I:GOTO 210
300  DATA 9,9-1,9,9,-1,9,9,-1,9,9,-1,9,9,-1,9,9,-1,2,16,23,38,-1,9,9,-1,
9,9,-1,9,9,-1,9,9,-1,9,9,-1,9,9,-1,9,15,-1,50
310  DATA 9,9,-1,9,9,-1,9,9,-1,9,9,-1,9,9,-1,9,9,-1,2,16,23,38,-1,9,9,-1,9
,9,-1,9,9,-1,9,9,-1,9,.9,-1,9,9,-1,9,9,-1,50
```

程序 2 如下:

```
5  REM  PROGRAM  103.2
10  POKE  103,1: POKE  104, 96: POKE  24576,0
20  PRINT  CHR $  (4)"RUN  TEXT - 2"
```

104. 如何在低分辨率第二页绘制低分辨率图形

如同不能直接在文本第二页写字符一样，通常也不能直接在低分辨率第二页绘图。为了能在低分辨率第二页绘图，也应将程序区移至其它地方。移动的方法同问题"如何在文本第二页写字符"中叙述的方法一样，程序区移动后，也不能象在文本第二页直接写字符那样，直接在低分辨率第二页绘图。需采用在低分辨率第一页绘完图后，再迁移到低分辨率第二页的间接绘图法。举例如下:

先将程序 1 以名字 GR-2 存入磁盘中。然后再运行程序 2 即可在低分辨率第二页绘制 80 朵彩色小花。

程序 1 如下:

```
5  REM PROGRAM 104.1
10  GR : POKE - 16302,0:COLOR = 7
20  FOR Y = 0 TO 47
30  HLIN 0, 39 AT Y
40  NEXT Y
45  COLOR = 9
50  FOR Y = 0 TO 39
60  HLIN 0, Y AT Y
70  NEXT Y
80  POKE 768, 160: POKE 769,0
```

```
90  POKE 770, 76: POKE 771,44
100  POKE 772,25
110  POKE 60,0: POKE 61,4
120  POKE 62,255:POKE 63,7
130  POKE 66, 0: POKE 67,8
140  CALL 768
150  GR : GET A $
160  POKE - 16299,0: POKE - 16302,0
```

程序 2 如下:

```
5  REM  PROGRAM 104.2
10  POKE 103,1: POKE 104,96: POKE 24576,0
20  PRINT  CHR $ (4)"RUN GR -2"
```

105. 如何使高分辨率图形反相显示

将高分辨率图形反相显示的方法是: 用 255（ $FF)减去存放高分辨率图形数据的内存单元中的数值，再存放在该内存单元中。

用 BASIC 语言编写的程序如下:

```
5  REM PROGRAM 105.1
10  PRINT CHR $ (4)"BLOAD HPLOT-5"
20  POKE - 16304,0: POKE - 16297,0
30  GET A $
40  FOR P = 8192 TO 16383
50  POKE P,255 - PEEK (P)
60  NEXT P
```

该程序运行后，图象的反相过程较慢。如果要加快图的反相速度可使用机器语言。机器语言程序如下:

```
* 300L
0300-  A9  00      LDA   #$00
0302-  85  01      STA   $01
0304-  A9  20      LDA   #$20
0306-  85  02      STA   $02
0308-  A5  00      LDA   $00
030A-  C9  02      CMP   #$02
```

```
030E-  A9 40     LDA   #$40
0310-  85 02     STA   $02
0312-  A0 00     LDY   #$00
0314-  A2 20     LDX   #$20
0316-  B1 01     LDA   ($01),Y
0318-  0A        ASL
0319-  49 FF     EOR   #$FF

031B-  6A        ROR
031C-  91 01     STA   ($01),Y
031E-  E6 01     INC   $01
0320-  D0 F4     BNE   $0316
0322-  E6 02     INC   $02
0324-  CA        DEX
0325-  D0 EF     BNE   $0316
0327-  60        RTS
```

如果要使高分辨率第一页图形反相，可用 POKE 0,1：CALL 768 命令。如果要使高分辨率第二页图形反相可用 POKE 0，2：CALL768 命令举例如下：

```
5  REM PROGRAM 105.2
10  D$ = CHR$ (4)
20  PRINT D$;"BLOAD HPLOT-N"
30  PRINT D$;"BLOAD HPLOT-5"
40  POKE - 16304,0: POKE - 16297,0
50  GET A$
60  POKE 0,1: CALL 768
```

106. 如何水平移动高分辨率图形

运行下面各程序可使高分辨率图形水平向左或向右移动，以及使高分辨率图形水平移位卷绕。

一．使高分辨率图形水平移动

```
5  REM PROGRAM 106.1
10  HOME: VTAB 10: HTAB 10
20  INPUT "LEFT OR RIGHT:";A$
25  PRINT : HTAB 10: INPUT "NAME:";F$
```

```
30  PRINT : HTAB 10: INPUT "N=";N
35  PRINT CHR$ (4);"BLOAD ";F$
40  POKE -16304,0: POKE - 16297,0
50  GET A$ : HGR2
60  IF A$ = "R" THEN A = 39 - N:B = 0:C = - 1: GOTO 80
70  A = N:B = 39:C = 1:N = - N
80  FOR I = 0 TO 80 STEP 40
90  FOR J = I + 8192 TO 16383 STEP 128
100  FOR K = J + A TO J + B STEP C
110  POKE 8192 + K + N, PEEK (K)
120  NEXT K,J,I
```

程序中，A$ ="R" 时为右移，A$ ="L" 时为左移；F$ 为磁盘中图形文件的名字；N 是水平移动量 (1———40)。

二. 使高分辨率图形水平右移卷绕

```
5   REM PROGRAM 106.2
10  HOME : VYAB 10: HTAB 10: INPUT "N=";N
20  PRINT CHR$ (4);"BLOAD HPLOT-5"
30  POKE - 16304,0: POKE - 16297,0
40  GET A$ : HGR2
50  FOR I = 0 TO 80 STEP 40
60  FOR J = I + 8192 TO 16383 STEP 128
70  FOR K = 0 TO 39
80  P1 = J + K
90  L = K + N: IF L > 39 THEN L = L - 40
100  P2 = J + L + 8192
110  POKE P2, PEEK (P1)
120  NEXT K,J,I
```

程序中，N 为水平移动量(1——40)。

三.使高分辨率图形水平左移卷绕

运行下面的 BASIC 程序后，输入要处理的图形文件名(赋给 F$)，机器调出图形数据后再将一个名字为" HPLOT-L "的机器语言程序调入主机内存单元中。以后，图形缓慢向左移动卷绕,当按下任一键，图形停止移动。

BASIC 程序和机器语言程序如下:

```
5   REM  PROGRAM  106.3
10  HOME : VTAB  10: HTAB  10: INPUT "NAME:";F$
20  PRINT   CHR$  (4);"BLOAD";F$;",A$4000"
30  PRINT   CHR$  (4);"BLOAD HPLOT-L"
```

```
35  POKE - 16304,0: POKE - 16297,0
40  CALL 768
50  A =  PEEK ( - 16384)
60  IF A > 128 THEN END
70  GOTO 40
```

```
* 300.360
0300-  A9 40 85 FB A9 00 85 1A
0308-  A9 28 85 1B A9 50 85 1C
0310-  A9 80 85 1D A9 A8 85 1E
0318-  A9 D0 85 1F A2 06 B5 19
0320-  85 FA 8A 48 A0 00 B1 FA
0328-  AA A0 28 88 8A 6A 6A 29
0330-  80 85 19 B1 FA AA 29 7F
0338-  05 19 4A 85 19 8A 29 80
0340-  05 19 91 FA 98 D0 E4 68
0348-  AA CA D0 D2 E6 FB A5 FB
0350-  C9 60 90 C8 2C 50 C0 2C
0358-  52 C0 2C 57 C0 2C 55 C0
0360-  60
* 3D0G
```

107. 如何使高分辨率图形上下颠倒

运行下面程序可以获得上下颠倒高分辨率图形。

一. BASIC 程序

```
5   REM PROGRAM 107.1
10  D$ = CHR$ (4)
20  PRINT D$;"BLOAD HPLOT-5"
30  POKE - 16304,0: POKE - 16297,0
40  GET A$: HGR2
50  FOR I = 0 TO 2
60  FOR J = 0 TO 7
70  FOR K = 0 TO 7
80  FOR L = 0 TO 39
90  IF I = 1 AND J = 4 THEN END
100  P1 = 8192 + 40 * I + 128 * J + 1024 * K + L
110  P2 = 16384 + 40 * (2 - I) + 128 * (7 - J) + 1024 * (7 - K) + L
120  POKE P2, PEEK (P1): POKE P1 + 8192, PEEK (P2 - 8192)
130  NEXT L,K,J,I
```

该程序运行后，可在高分辨率第二页获得将高分辨率第一页图形上下颠倒的图形。程序中 P1 与 P2 分别为上下对称的内存单元地址。如果要在高分辨率第一页获得该页原图形上下颠倒的图形，可将上面这个程序稍加修改，修改后的 BASIC 程序如下。

```
5   REM  PROGRAM  107.2
10   D$  = CHR$ (4)
20   PRINT D$;"BLOAD HPLOT-5"
30   POKE - 16304,0: POKE - 16297,0
40   GET  A$
50   FOR  I = 0  TO  2
60   FOR  J = 0  TO  7
70   FOR  K = 0  TO  7
80   FOR  L = 0  TO  39
90   IF  I = 1 AND J = 4 THEN END
100   P1 = 8192 + 40 * I + 128 * J + 1024 * K + L
110   P2 = 8192 + 40 * (2 - I) + 128 * (7 - J) + 1024 * (7 - K) + L
120   N = PEEK(P2): POKE  P2, PEEK(P1): POKE  P1,N
130   NEXT L,K,J,I
```

如果要获得上半边图形的倒影图形，可将上边这个程序中的第 120 语句改为:

```
120   POKE P2 ,PEEK(P1)
```

二. 机器语言程序

机器语言子程序如下:

```
0300-  8D  50  C0      STA   $C050
0303-  8D  52  C0      STA   $CO52
0306-  8D  54  C0      STA   $CO54
0309-  8D  57  C0      STA   $C057
030C-  A2  00          LDX   #$00
030E-  BD  55  03      LDA   $0355,X
0311-  85  05          STA   $05
0313-  E8              INX
0314-  BD  55  03      LDA   $0355,X
0317-  85  04          STA   $04
0319-  E8              INX
031A-  BD  55  03      LDA   $0355,X
031D-  18              CLC
031E-  69  1C          ADC   #$1C
0320-  85  07          STA   $07
0322-  E8              INX
0323-  BD  55  03      LDA   $0355,X
0326-  85  06          STA   $06
0328-  A9  08          LDA   #$08
032A-  85  08          STA   $08
032C-  A0  00          LDY   #$00
032E-  B1  04          LDA   ($04),Y
0330-  48              PHA
0331-  B1  06          LDA   ($06),Y
```

```
0333-  91  04          STA  ($04),Y
0335-  68              PLA
0336-  91  06          STA  ($06),Y
0338-  C8              INY
0339-  C0  28          CPY  #$28
033B-  D0  F1          BNE  $032E
033D-  A5  05          LDA  $05
033F-  18              CLC
0340-  69  04          ADC  #$04
0342-  85  05          STA  $05
0344-  A5  07          LDA  $07
0346-  38              SEC
0347-  E9  04          SBC  #$04
0349-  85  07          STA  $07
034B-  C6  08          DEC  $083
034D-  D0  DD          BNE  $032C
034F-  E8              INX
0350-  E0  30          CPX  #$30
0352-  D0  BA          BNE  $030E
0354-  60              RTS
```

高分辨率图形内存单元地址数据如下:

```
0355-  20  00  23
0358-  D0  20  80  23  50  21  00  22
0360-  D0  21  80  22  50  22  00  21
0368-  D0  22  80  21  50  23  00  20
0370-  D0  23  80  20  50  20  28  23
0378-  A8  20  A8  23  28  21  28  22
0380-  A8  21  A8  22  28
```

应用该机器语言程序进行高分辨率图形上下颠倒的实例如下:

```
5  REM PROGRAM 107.3     10  D$ = CHR$(4)
20  PRINT D$;"BLOAD D-W"
30  PRINT D$;"BLOAD HPLOT-5"
40  POKE -16304,0: POKE -16297,0
50  GET A$: HGR2
60  CALL 768
```

108. 如何使高分辨率图形左右颠倒

运行下面的程序可以获得左右颠倒的高分辨率图形。

一. BASIC 程序

```
5  REM PROGRAM 108.1
10  D$ = CHR$ (4)
20  PRINT D$ ;"BLOAD HPLOT -5"
30  POKE - 16304,0: POKE - 16297,0
40  GET A$ : HGR2
50  FOR I = 0 TO 2
60  FOR J = 0 TO 7
70  FOR K = 0 TO 7
80  FOR L = 0 TO 39
90  P1 = 8192 + 40 * I + 128 * J + 1024 * K + L6
100  A = PEEK (P1): GOSUB 200
110  P2 = P1 + 8192 +39 -2 * L
120  POKE P2, B:B = 0:C = 0
130  NEXT L, K, J, I
140  END
200  IF A > 128 THEN A= A -128:C = 128
210  FOR N = 0 TO 6
220  Q(N) = A - INT (A / 2) * 2:A = INT (A / 2)
230  B = B + Q(N) * 2 ∧(6 - N)
240  NEXT N
250  B = B +C: RETURN
```

该程序运行后，可在高分辨率第二页获得将高分辨率第一页图形左右颠倒的图形程序中 P1 与 P2 分别为左右对称的内存单元地址。由于颠倒的图形不仅要把每条扫描线上各单元（一个单元对应七个色点）自左往右的次序颠倒过来，而且还要把每个单元的七个色点次序颠倒过来。为此，程序中 200——250 语句构成一个颠倒内存单元中的七位数的子程序。如果要在高分辨率第一页获得该页原图形左右颠倒的图形，可将上面的程序稍加修改，修改后的程序如下:

```
5  REM PROGRAM 108.2
10  D$ = CHR$(4)
20  PRINT D$;"BLOAD HPLOT-5"
```

```
30  POKE - 16304,0: POKE - 16297,0
40  GET A $
50  FOR I = 0 TO 2
60  FOR J = 0 TO 7
70  FOR K = 0 TO 7
80  FOR L = 0 TO 19
90  P1 = 8192 + 40 * I + 128 * J + 1024 * K +L
100  A  = PEEK(P1): GOSUB 200
110  P2 = P1 + 39 - 2 * L
120  A = PEEK (P2): POKE P2,B:B = 0: C = 0:GOSUB 200 : POKE P1, B: B = 0 :C=0
130  NEXT L,K,J,I
140  END
200  IF A > 128 THEN A = A- 128: C = 128
210  FOR N = 0 TO 6
220  Q(N) = A - INT (A / 2) * 2: A = INT ( A / 2)
230  B = B + Q (N) * 2^(6-N)
240  NEXT N
250  B = B + C: RETURN
```

如果要获得左半边图形的镜象图形，可将上面的程序中第 120 语句改为:

```
120  POKE P2, B: B=0:C=0
```

另外，运行下面这个程序也可获得同样效果:

```
5  REM PROGRAM 108.3
10  D $ = CHR $ (4)
20  PRINT D $ ;"BLOAD HPLOT - 5"
30  PUKE - 16304,0: POKE - 16297,0
40  GET A $
50  FOR I = 0 TO 80 STEP 40
60  FOR J = 8192 + I TO 16383 STEP 128
70  FOR K = J TO J + 19
80  A = PEEK (K): GOSUB 200
90  P = J + 39 + J - K
100  POKE P,B:B = 0:C = 0
110  NEXT K,J,I
200  IF A > 128 THEN A = A - 128: C = 128
210  FOR N = 0 TO 6
220  Q(N) = A - INT (A / 2) * 2:A = INT (A / 2)
```

```
230  B = B+Q(N) * 2^(6-N)
240  NEXT N
250  B = B+C: RETURN
```

二. 机器语言程序

机器语言程序如下:

```
 * 300.378
0300-  A9  00  85  11  20  53  03  A0
0308-  00  B1  14  48  C0  27  F0  04
0310-  C8  4C  09  03  A0  00  68  A2
0318-  00  86  00  E8  0A  66  00  E
0320-  08  D0  F8  06  00  26  01  46
0328-  00  46  00  26  03  46  02  26
0330-  00  26  00  46  03  66  00  46
0338-  01  26  00  A5  00  91  14  C0
0340-  27  F0  04  C8  4C  16  03  A0
0348-  BF  C4  11  F0  05  E6  11  4C
0350-  04  03  60  A5  11  0A  0A  29
0358-  1C  85  15  A5  11  6A  6A  6A
0360-  6A  29  03  05  15  09  20  85
0368-  15  A5  11  6A  29  E0  85  14
0370-  6A  6A  29  18  05  14  85  14
0378-  60
```

应用机器语言程序进行高分辨率图形颠倒的实例如下:

```
5   REM PROGRAM 110.4
10  D$ =  CHR$ (4)
20  PRINT D$;"BLOAD L-R"
30  PRINT D$;"BLOAD HPLOT-5"
40  POKE-16304,0: POKE-16297,0
50  GET A$: CALL 768
```

109. 如何使屏幕上的字符自上而下或从左至右卷绕

一. 使文本屏幕上的字符自上而下卷绕

运行下面的程序即可实现

```
0300-  A5  20  18  65  21  8D  6D  03
0308-  CE  6D  03  A5  23  18  E9  01
0310-  20  C1  FB  AC  6D  03  B1  28
0318-  99  6E  03  88  C4  20  10  F6
0320-  A5  23  38  E9  01  48  AA  20
0328-  C1  FB  A5  28  85  00  A5  29
0330-  85  01  CA  8A  20  C1  FB  AC
0338-  6D  03  B1  28  91  00  88  C4
0340-  20  10  F7  68  38  E9  01  C5
0348-  22  D0  DA  AC  6D  03  A5  22
0350-  20  C1  FB  B9  6E  03  91  28
0358-  88  C4  20  10  F6  A9  A0  20
0360-  A8  FC  AD  00  C0  10  A4  A9
0368-  00  8D  10  C0  60
*
```

二. 使文本屏幕上字符从左至右的卷绕

运行下面的程序即可实现

```
0300-  A5  20  18  65  21  8D  61  03
0308-  CE  61  03  A6  23  CA  8A  20
0310-  C1  FB  AC  61  03  B1  28  9D
0318-  62  03  CA  E4  22  10  EF  AC
0320-  61  03  88  A6  23  CA  8A  48
0328-  20  C1  FB  B1  28  C8  91  28
0330-  88  68  38  E9  01  C5  22  10
0338-  EE  88  C4  20  10  E5  A6  23
0340-  CA  8A  20  C1  FB  A4  20  BD
0348-  62  03  91  28  CA  E4  22  10
0350-  F0  A9  BB  20  A8  FC  AD  00
0358-  C0  10  B0  A9  00  8D  10  C0
0360-  60
```

110. 如何实现幻灯式地连续显示图形

为了实现幻灯式地连续显示图形，可将各图形数据以名字为 PIC−1———PIC−N 存入一张或多张磁盘中。如果存放图形数据的磁盘不是一张,则需用两台驱动器。

然后，运行下面的程序可实现两页交替装载交替显示，从而达到幻灯式连续显示图形的效果。而且在交替显示时，画面整幅出现。使用下面这个程序时应注意：每张磁盘必须存放 10 个高分辨率图形数据，而最后一张磁盘存放的高分辨率图形数据不大于 10 个即可。

```
5   REM PROGRAM 110.1
10  HOME: VTAB 10: HTAB 10
20  INPUT "N=";N
30  D$ = CHR$ (4): HGR2
40  FOR I = 1 TO N
50  S = INT (I / 11)+1
60  IF S / 2 = INT (S / 2) THEN D = 2: GOTO 80
70  D = 1
80  PRINT D$;"BLOAD PIC-";I;",A$2000,D";D
90  POKE -16300,0:I = I+1
100  IF I > N THEN 140
110  PRINT D$;"BLOAD PIC-";I;",A$4000,D";D
120  POKE -16299,0
130  NEXT I
```

如果显示的是低分辨率图形或是文本文字，则需运行下面程序 PRG−1：

```
5  REM PROGRAM 110.2
```

```
10  POKE 103,1: POKE 104,96: POKE 24576,0
20  PRINT  CHR $ (4);"RUN GR-N"
```

程序 PRG-1 调用的程序 GR-N 如下:

```
5  REM PROGRAM 110.3
10  HOME : VTAB 10: HTAB 10
20  INPUT "N=";N
30  D $ = CHR $ (4): GR : POKE - 16302,0: POKE - 16299,0
40  FOR I = 1 TO N
50  S = INT (I / 90) + 1
60  IF S / 2 = INT (S / 2) THEN D = 2: GOTO 80
70  D = 1
80  PRINT D $ ;"BLOAD PIC-";I;",A $ 400,D";D
90  POKE - 16300,0: I = I + 1
100  IF I > N THEN 140
110  PRINT D $ ;"BLOAD PIC-";I;",A $ 800,D";D
120  POKE - 16299,0
130  NEXT I
140  END
```

111. 如何使图形从中间向上下同时缓慢延伸显示

运行下面的程序可使图形由中间向上向下同时缓慢延伸显示出来。将下面的程序进行一些改动,便可使图形由中间向左向右同时缓慢延伸显示,也可使图形由上下向中间缓慢延伸显示或由左右向中间缓慢延伸显示。

一. BASIC 程序法

```
5  REM PROGRAM 111.1
10  HOME : VTAB 10: HTAB 10
20  INPUT "PIC NAM:";F $
30  PRINT CHR $ (4);"BLOAD ";F $ ;",A $ 2000"
40  POKE - 16304,0: POKE - 16297, 0
50  GET A $ : HGR2 :P = 8192 + 40
60  FOR A = 3 TO 0 STEP - 1
70  FOR B = 0 TO 7
80  FOR C = 0 TO 39
90  P1 = P + 128 * A + 1024 * (7 - B) + C
100  P2 = P + 128 * (7 - A) + 1024 * B + C
110  POKE P1 + 8192, PEEK (P1): POKE P2 + 8192, PEEK(P2)
```

```
120  NEXT C,B,A
130  FOR A = 7 TO 0 STEP - 1
140  FOR B = 0 TO 7
150  FOR C = 0 TO 39
160  P1 = 8192 + 128 * A + 1024 * (7 - B) + C
170  P2 = 8192 + 80 + 128 * (7 - A) + 1024 * B + C
180  POKE P1 + 8192, PEEK (P1): POKE P2 + 8192, PEEK (P2)
190  NEXT C,B,A
200  END
```

二. 机器语言程序法

运行下面的 BASIC 程序，用它调用存放在以 $300 为起始地址的内存区域中的机器语言，便可实现图形由中间向上向下同时缓慢延伸显示。

BASIC 程序如下:

```
5   REM PROGRAM 111.2
10  D$ = CHR$ (4)
20  HOME: VTAB 10: HTAB 10 : INPUT "PIC NAME:";F$
30  PRINT D$;"BLOAD";F$;",A$2000"
40  CALL 768
```

机器语言程序如下:

```
0300- 48 8A 48 98 48 20 D8 F3      0338- B1 26 91 FE E6 26 90 02
0308- A2 20 86 E6 A2 5F A0 00      0340- E6 27 CA D0 E9 A5 00 18
0310- 86 00 84 01 A4 01 98 18      0348- 65 02 85 04 C6 03 D0 D5
0318- 0A A8 C8 84 02 A2 02 86      0350- A9 30 20 A8 FC C6 00 E6
0320- 03 A5 00 85 04 A5 04 20      0358- 01 A5 01 C9 60 D0 B5 68
0328- 11 F4 A2 28 A0 00 A5 26      0360- A8 68 AA 68 60
0330- 85 FE A5 27 69 20 85 FF
```

112. 如何使图形由中间向四周同时延伸显示

运行下面的 BASIC 程序，并通过它执行存放在 $8000—$80C4 内存单元中的机器语言程序，可实现将图象由中间向四周同时延伸显示。机器语言程序由三部分组成，$8000——$8079 存放主程序， $807A——$809E 及 $809F——$80C4 存放两个子程序。改变 $806E 内存单元中的数据，可改变显示速度。

程序如下:

```
5  REM PROGRAM 112.1
```

```
10  HOME :VTAB 10: HTAB 10
20  INPUT "PIC NAME:";F $
30  PRINT CHR $ (4);"BLOAD PIC-D"
40  PRINT CHR $ (4);"BLOAD ";F $ ",A $ 4000"
50  HGR : POKE - 16302,0
60  CALL 32768
```

```
*8000.80C4
8000- A9 5A 85 02 8D 7B 80 A9
8008- 66 85 03 8D 9B 80 A9 12
8010- 85 04 8D 82 80 A9 16 85
8018- 05 8D 96 80 20 7A 80 A2
8020- 00 86 06 A5 04 18 69 FF
8028- 8D 82 80 8D 96 80 20 7A
8030- 80 A5 05 8D 82 80 8D 42
8038- 03 20 7A 80 C6 04 E6 05
8040- A5 04 8D 82 80 A5 05 8D
8048- 96 80 A5 02 8D 98 80 69
8050- FA 85 02 8D 7B 80 20 7A
8058- 80 A5 03 8D 7B 80 69 04
8060- 85 03 8D 9B 80 20 7A 80
8068- A5 02 8D 7B 80 A9 20 20
8070- A8 FC A6 06 E8 E0 12 90
8078- A8 60 A2 00 86 11 20 9F
8080- 80 A0 00 A5 14 85 00 A5
8088- 15 85 01 18 69 20 85 15
8090- B1 14 91 00 C8 C0 28 90
8098- F7 E8 E0 C0 90 DE 60 A5
80A0- 11 0A 0A 29 1C 85 15 A5
80A8- 11 6A 6A 6A 6A 29 03 05
80B0- 15 09 20 85 15 A5 11 6A
80B8- 29 E0 85 14 6A 6A 29 18
80C0- 05 14 85 14 60
```

113. 如何使画面象瀑布倾泻似地显示

将存放在 $8000——$80CE 内存单元的机器语言子程序以名字"PIC-DD-1"存入磁盘，然后用下面的 BASIC 程序调用它，即可实现画面象瀑布倾泻似地显示。如果将 $8048 内存单元中的数改为 $B7，则复杂画面的显示效果更有趣。

```
*8000.80CE
8000- A9 A1 85 15 EA EA EA A9
8008- BF 85 15 C6 15 D0 01 60
8010- A9 01 85 17 A5 15 85 16
8018- 85 18 85 10 20 C9 80 A5
8020- 11 85 13 A5 12 A8 69 20
8028- 85 14 A5 16 85 10 20 C9
8030- 80 20 AC 80 C6 18 C6 16
8038- F0 D1 A6 17 A5 16 85 10
8040- 86 19 20 C9 80 A6 19 20
8048- B8 80 C6 16 F0 EA CA D0
8050- EB E6 17 A9 0C A0 0A A5
8058- 18 4C 1A 80 A5 10 29 07
8060- 0A 0A 18 69 20 85 12 A5
8068- 10 29 38 85 11 6A 6A 6A
8070- 6A 29 0F 18 65 12 85 12
8078- A5 11 0A 0A 0A 0A 85 11
8080- A5 10 6A 6A 6A 6A 6A 6A
8088- 29 03 AA A9 00 18 E0 03
8090- 90 02 69 27 E0 02 90 02
8098- 69 27 E0 01 90 02 69 27
80A0- 18 65 11 85 11 A9 00 65
80A8- 12 85 12 60 A0 27 B1 13
```

```
80B0-  91  11  88  C0  FF  D0  F7  60
80B8-  A0  27  A9  00  91  11  88  A9
80C0-  00  91  11  88  C0  FF  D0  F2
80C8-  60  C6  10  20  5C  80  60
```

```
5  REM PROGRAM 113.1
10  HOME: VTAB 10: HTAB 10
20  INPUT "PIC NAME:"; F $
30  PRINT CHR $ (4);"BLOAD PIC-DD-1"
40  PRINT CHR $ (4);"BLOAD";F $ ;,A $ 4000"
50  HGR : POKE - 16302,0
60  CALL 32768
```

114. 如何使图形呈流水状地显示

运行下面的 BASIC 程序，输入磁盘中图形数据的文件名即可使图形呈流水状显示出来。BASIC 程序调用两个机器语言程序，SB-1 程序由 4 个小程序和地址数据组成：$ A00—— $ A24、$ A25—— $ A50、$ A51—— $ AB6、$ AB7—— $ B1F 为 4 个小子程序，$ B20—— $ C9F 为地址数据。SB-2 程序存放在 $ 1000—— $ 1450 内存区域，它完成图形从上至下的注入过程。

BASIC 程序如下：

```
5  REM PROGRAM 114.1
10  D $  = CHR $ (4): HOME
20  PRINT D $ ;"BLOAD SB-1"
30  PRINT D $ ;"BLOAD SB-2"
40  VTAB 10: HTAB 10: INPUT "PIC NAME:";F $
50  PRINT D $ ;"BLOAD";F $ ;",A $ 4000"
60  HGR
70  CALL 2560
```

SB-1 程序如下：

```
0A00-  20  25  0A  A9  18  85  01  85
0A08-  00  20  51  0A  A9  09  85  01
0A10-  20  51  0A  18  A5  09  69  30
0A18-  85  09  A5  0A  69  00  85  0A
0A20-  C6  00  D0  E8  60  A9  20  85
0A28-  06  AA  A9  00  85  05  A8  91
0A30-  05  C8  D0  FB  E6  06  CA  D0
0A38-  F6  8D  50  C0  8D  52  C0  8D
0A40-  57  C0  85  09  A9  10  85  0A
0A48-  A9  00  85  02  A9  17  85  04
0A50-  60  A5  00  0A  0A  0A  AA  CA
0A58-  BD  20  +0B  85  05  BD  E0  0B
0A60-  85  06  A2  08  A0  02  B1  05
0A68-  88  88  91  05  C8  C8  C8  C0
0A70-  14  D0  F3  A0  25  B1  05  C8
0A78-  C8  91  05  88  88  88  C0  13
0A80-  D0  F3  38  A5  06  E9  04  85
0A88-  06  CA  D0  D8  A0  13  20  B7
0A90-  0A  C8  20  B7  0A  A0  12  20
0A98-  B7  0A  A0  15  20  B7  0A  20
```

```
0AA0- BA 0A A9 00 85 0B A9 10
0AA8- 85 0C C6 0B D0 FC C6 0C
0AB0- D0 F8 C6 01 D0 9B 60 6C
0AB8- 09 00 A5 04 0A 0A 0A AA
0AC0- BD 20 0B 85 05 BD E0 0B
0AC8- 49 60 85 06 A9 00 85 07
0AD0- A9 20 85 08 38 A9 27 E5
0AD8- 02 85 03 A2 08 A4 02 B1
0AE0- 05 A0 12 91 07 A4 02 B1
0AE8- 05 A0 15 91 07 A4 02 C8

0AF0- B1 05 A0 13 91 07 A4 03
0AF8- 88 B1 05 A0 14 91 07 18
0B00- A5 06 69 04 85 06 A5 08
0B08- 69 04 85 08 CA D0 CE E6
0B10- 02 E6 02 A5 02 C9 14 D0
0B18- 06 A9 00 85 02 C6 04 60
0B20- 00 00 00 00 00 00 00 00
0B28- 80 80 80 80 80 80 80 80
0B30- 00 00 00 00 00 00 00 00
0B38- 80 80 80 80 80 80 80 80
```

```
0B40- 00 00 00 00 00 00 00 00
0B48- 80 80 80 80 80 80 80 80
0B50- 00 00 00 00 00 00 00 00
0B58- 80 80 80 80 80 80 80 80
0B60- 28 28 28 28 28 28 28 28
0B68- A8 A8 A8 A8 A8 A8 A8 A8
0BA0- 50 50 50 50 50 50 50 50
0BA8- D0 D0 D0 D0 D0 D0 D0 D0
0BB0- 50 50 50 50 50 50 50 50
0BB8- D0 D0 D0 D0 D0 D0 D0 D0

0B70- 28 28 28 28 28 28 28 28
0B78- A8 A8 A8 A8 A8 A8 A8 A8
0B80- 28 28 28 28 28 28 28 28
0B88- A8 A8 A8 A8 A8 A8 A8 A8
0B90- 28 28 28 28 28 28 28 28
0B98- A8 A8 A8 A8 A8 A8 A8 A8
0BC0- 50 50 50 50 50 50 50 50
0BC8- D0 D0 D0 D0 D0 D0 D0 D0
0BD0- 50 50 50 50 50 50 50 50
0BD8- D0 D0 D0 D0 D0 D0 D0 D0
```

```
0BE0- 20 24 28 2C 30 34 38 3C
0BE8- 20 24 28 2C 30 34 38 3C
0BF0- 21 25 29 2D 31 35 39 3D
0BF8- 21 25 29 2D 31 35 39 3D
0C00- 22 26 2A 2E 32 36 3A 3E
0C08- 22 26 2A 2E 32 36 3A 3E
0C10- 23 27 2B 2F 33 37 3B 3F
0C18- 23 27 2B 2F 33 37 3B 3F
0C20- 20 24 28 2C 30 34 38 3C
0C28- 20 24 28 2C 30 34 38 3C
0C30- 21 25 29 2D 31 35 39 3D
0C38- 21 25 29 2D 31 35 39 3D

0C40- 22 26 2A 2E 32 36 3A 3E
0C48- 22 26 2A 2E 32 36 3A 3E
0C50- 23 27 2B 2F 33 37 3B 3F
0C58- 23 27 2B 2F 33 37 3B 3F
0C60- 20 24 28 2C 30 34 38 3C
0C68- 20 24 28 2C 30 34 38 3C
0C70- 21 25 29 2D 31 35 39 3D
0C78- 21 25 29 2D 31 35 39 3D
0C80- 22 26 2A 2E 32 36 3A 3E
0C88- 22 26 2A 2E 32 36 3A 3E
0C90- 23 27 2B 2F 33 37 3B 3F
0C98- 23 27 2B 2F 33 37 3B 3F
```

SB-2 程序如下:

```
1000- B9 50 3F 99 D0 3F B9 50
1008- 3B 99 D0 3B B9 50 37 99
1010- D0 37 B9 50 33 99 D0 33
1018- B9 50 2F 99 D0 2F B9 50
1020- 2B 99 D0 2B B9 50 27 99
1028- D0 27 B9 50 23 99 D0 23
1030- B9 D0 3E 99 50 3F B9 D0
1038- 3A 99 50 3B B9 D0 36 99
1040- 50 37 B9 D0 32 99 50 33
1048- B9 D0 2E 99 50 2F B9 D0
```

```
1050- 2A 99 50 2B B9 D0 26 99
1058- 50 27 B9 D0 22 99 50 23
1060- B9 50 3E 99 D0 3E B9 50
1068- 3A 99 D0 3A B9 50 36 99
1070- D0 36 B9 50 32 99 D0 32
1078- B9 50 2E 99 D0 2E B9 50
1080- 2A 99 D0 2A B9 50 26 99
1088- D0 26 B9 50 22 99 D0 22
1090- B9 D0 3D 99 50 3E B9 D0
1098- 39 99 50 3A B9 D0 35 99
```

```
10A0- 50 36 B9 D0 31 99 50 32
10A8- B9 D0 2D 99 50 2E B9 D0
10B0- 29 99 50 2A B9 D0 25 99
10B8- 50 26 B9 D0 21 99 50 22
10C0- B9 50 3D 99 D0 3D B9 50
10C8- 39 99 D0 39 B9 50 35 99
10D0- D0 35 B9 50 31 99 D0 31
10D8- B9 50 2D 99 D0 2D B9 50
10E0- 29 99 D0 29 B9 50 25 99
10E8- D0 25 B9 50 21 99 D0 21
```

```
10F0- B9 D0 3C 99 50 3D B9 D0
10F8- 38 99 50 39 B9 D0 34 99
1100- 50 35 B9 D0 30 99 50 31
1108- B9 D0 2C 99 50 2D B9 D0
1110- 28 99 50 29 B9 D0 24 99
1118- 50 25 B9 D0 20 99 50 21
1120- B9 50 3C 99 D0 3C B9 50
1128- 38 99 D0 38 B9 50 34 99
1130- D0 34 B9 50 30 99 D0 30
1138- B9 50 2C 99 D0 2C B9 50
```

```
1140- 28 99 D0 28 B9 50 24 99
1148- D0 24 B9 50 20 99 D0 20
1150- B9 A8 3F 99 50 3C B9 A8
1158- 3B 99 50 38 B9 A8 37 99
1160- 50 34 B9 A8 33 99 50 30
1168- B9 A8 2F 99 50 2C B9 A8
1170- 2B 99 50 28 B9 A8 27 99
1178- 50 24 B9 A8 23 99 50 20
1180- B9 28 3F 99 A8 3F B9 28
1188- 3B 99 A8 3B B9 28 27 99
```

```
1190- A8 37 B9 28 33 99 A8 33
1198- B9 28 2F 99 A8 2F B9 28
11A0- 2B 99 A8 2B B9 28 27 99
11A8- A8 27 B9 28 23 99 A8 23
11B0- B9 A8 3E 99 28 3F B9 A8
11B8- 3A 99 28 3B B9 A8 36 99
11C0- 28 37 B9 A8 32 99 28 33
11C8- B9 A8 2E 99 28 2F B9 A8
11D0- 2A 99 28 2B B9 A8 26 99
11D8- 28 27 B9 A8 22 99 28 23
```

```
11E0- B9 28 3E 99 A8 3E B9 28
11E8- 3A 99 A8 3A B9 28 36 99
11F0- A8 36 B9 28 32 99 A8 32
```

```
11F8- B9 28 2E 99 A8 2E B9 28
1200- 2A 99 A8 2A B9 28 26 99
1208- A8 26 B9 28 22 99 A8 22
```

```
1210– B9 A8 3D 99 28 3E B9 A8
1218– 39 99 28 3A B9 A8 35 99
1220– 28 36 B9 A8 31 99 28 32
1228– B9 A8 2D 99 28 2E B9 AB
1230– 29 99 28 2A B9 A8 25 99
1238– 28 26 B9 A8 21 99 28 22
1240– B9 28 3D 99 A8 3D B9 28

1248– .39 99 A8 39 B9 28 35 99
1250– A8 35 B9 28 31 99 A8 31
1258– B9 28 2D 99 A8 2D B9 28
1260– 29 99 A8 29 B9 28 25 99
1268– A8 25 B9 28 21 99 A8 21
1270– B9 A8 3C 99 28 3D B9 A8
1278– 38 99 28 39 B9 A8 34 99

1280– 28 35 B9 A8 30 99 28 31
1288– B9 A8 2C 99 28 2D B9 A8
1290– 28 99 28 29 B9 A8 24 99
1298– 28 25 B9 A8 20 99 28 21
12A0– B9 28 3C 99 AB 3C B9 28
12AB– 38 99 A8 38 B9 28 34 99
12B0– A8 34 B9 28 30 99 A8 30
12B8– B9 28 2C 99 A8 2C B9 28
12C0– 28 99 A8 28 B9 28 24 99
12C8– A8 24 B9 28 20 99 A8 20

12D0– B9 80 3F 99 28 3C B9 80
12D8– 3B 99 28 38 B9 80 37 99
12E0– 28 34 B9 80 33 99 28 30
12E8– B9 80 2F 99 28 2C B9 80
12F0– 2B 99 28 28 B9 80 27 99
12F8– 28 24 B9 80 23 99 28 20
1300– B9 00 3F 99 80 3F B9 00
1308– 3B 99 80 3B B9 00 37 99
1310– 80 37 B9 00 33 99 80 33
1318– B9 00 2F 99 80 2F B9 00

130– 2B 99 80 2B B9 00 27 99
1328– 80 27 B9 00 23 99 80 23
1330– B9 80 3E 99 00 3F B9 80
1338– 3A 99 00 3B B9 80 36 99
1340– 00 37 B9 80 32 99 00 33
1348– B9 80 2E 99 00 2F B9 80
1350– 2A 99 00 2B B9 80 26 99
1358– 00 27 B9 80 22 99 00 23
1360– B9 00 3E 99 80 3E B9 00
1368– 3A 99 80 3A B9 00 36 99

1370– 80 36 B9 00 32 99 80 32
1378– B9 00 2E 99 80 2E B9 00
1380– 2A 99 80 2A B9 00 26 99
1388– 80 26 B9 00 22 99 80 22
1390– B9 80 3D 99 00 3E B9 80
1398– 39 99 00 3A B9 80 35 99
13A0– 00 36 B9 80 31 99 00 32
13A8– B9 80 2D 99 00 2E B9 80
13B0– 29 99 00 2A B9 80 25 99
13B8– 00 26 B9 80 21 99 00 22

13C0– B9 00 3D 99 80 3D B9 00
13C8– 39 99 80 39 B9 00 35 99
13D0– 80 35 B9 00 31 99 80 31
13D8– B9 00 2D 99 80 2D B9 00
13E0– –29 99 80 29 B9 00 25 99
13E8– 80 25 B9 00 21 99 80 21
13F0– B9 80 3C 99 00 3D B9 80

13F8– 38 99 00 39 B9 80 34 99
1400– 00 35 B9 80 30 99 00 31
1408– B9 80 2C 99 00 2D B9 80
1410– 28 99 00 29 B9 80 24 99
1418– 00 25 B9 80 20 99 00 21
1420– B9 00 3C 99 80 3C B9 00
1428– 38 99 80 38 B9 00 34 99
```

```
1430-  80  34  B9  00  30  99  80  30        1448-  80  24  B9  00  20  99  80  20
1438-  B9  00  2C  99  80  2C  B9  00        1450-  60
1440-  28  99  80  28  B9  00  24  99
```

115. 如何将高分辨率图形信息压缩

如果要用 BSAVE f, A $ 2000, L $ 1FFF 或 BSAVE f, A $ 4000,L $ 1FFF 将高分辨率图形数据存入磁盘，将占据磁盘 34 扇区。采用下述方法可将图形数据压缩。

一. BASIC 语言方法

运行下面的"PIC-I"BASIC 程序，可将高分辨率图形数据压缩后存入磁盘中。运行下面的"PIC-U" BASIC 程序可将压缩后存入磁盘的数据进行处理，使其高分辨率图形显示出来。在使用这两个程序时应输入要处理的图形数据文件名。

"PIC-I" BASIC 程序如下：

```
5   REM PROGRAM 115.1
10  D$ = CHR$ (4): HOME : VTAB 10 : HTAB 10
20  INPUT "PIC NAME:";F$
30  PRINT CHR$ (4);"BLOAD";F$;",A$2000"
40  DA = 16385
50  FOR N1 = 0 TO 2
60  FOR N2 = 0 TO 7
70  FOR N3 = 0 TO 7
80  FOR X = 0 TO 39
90  P = 8192 + 40 * N1 + 128 * N2 + 1024 * N3 + X :D = PEEK (P)
100 IF D > 127 THEN D = D - 128
110 IF D = 0 THEN M = M + 1: GOTO 180
120 IF M > 3 THEN 220
130 IF I = 0 THEN 230
140 IF I > 251 THEN I = 0 : GOTO 230
150 IF M = 0 THEN 170
160 FOR J = 1 TO M:DA = DA + 1: POKE DA,0: I = I + 1: POKE NA,I: NEXT J
170 DA = DA + 1: POKE DA,D: I = I + 1: POKE NA,I:M = 0
180 NEXT X:Y = Y + 1: NEXT N3: NEXT N2: NEXT N1
190 L = DA - 16383:HL = INT (L / 256):LL = L - HL * 256: POKE 16384,
HL: POKE 16385,LL
200 PRINT D$;"BSAVE";F$;"F,A$4000,L";L
210 END
220 XB = X:YB = Y: GOSUB 270: GOTO 170
230 IF M = 0 THEN 220
240 IF X < M THEN 260
```

```
250 XD = X − M:YB = Y: GOSUB 270: GOTO 160
260 XB = X + 40 − M: YB = Y − 1: GOSUB 270: GOTO 160
270 YA = DA + 1:XA = YA + 1:NA = YA + 2:DA = NA: POKE YA,YB: POKE XA,XB: I = 0
280  RETURN
```

"PIC−U" BASIC 程序如下:

```
5   REM PROGRAM 115.2
10  D$ = CHR$ (4): HOME : VTAB 10 : HTAB 10
20  INPUT " PIC NAME:";F$
30  PRINT D$;"BLOAD";F$;",A$6000"
40  F = 8192: HGR : POKE − 16302,0
50  DA = 24577:C = 2:N = PEEK (24576) * 256 + PEEK (24577)
60  YA = DA + 1: XA = YA + 1:NA = YA + 2: DA = NA:Y = PEEK (YA): X =
    PEEK (XA): I = PEEK (NA): C = C + I + 3
70  N1 = INT (Y / 64):N2 = Y − N1 * 64:N3 = INT (N2 / 8):N4 = N2 − N3 * 8:
   A = F + N1 * 40 + N3 * 128 + N4 * 1024 + X − 1
80  DA = DA + 1:D = PEEK (DA): A = A + 1: POKE A,D: I = I − 1
90  IF I = 0 THEN 120
100  IF X = 39 THEN X = 0 :Y = Y + 1: GOTO 70
110 X = X + 1: GOTO 80
120  IF C = N THEN END
130  GOTO 60
```

二. 机器语言方法

运行下面的程序可以压缩高分辨率图形数据:

```
5   REM PROGRAM 115.3
10  D$ CHR$ (4): HOME
20  PRINT D$;"BLOAD JB−1"
30  VTAB 10: HTAB 10: INPUT "PIC NAME:";F$
40  PRINT D$;"BLOAD ";F$;",A$2000"
50  CALL 327678
60  L = PEEK (220) + 256 * PEEK (221) − 16384
70  PRINT D$;"BSAVE";F$;"F,A$4000,L";L
```

"JB−1" 机器语言程序如下:

```
8000−  A9  1F  85  DF  A9  40  85  DD
8008−  A9  00  85  DC  A0  00  A2  01
8010−  A9  FF  85  DA  85  DE  20  5E
8018−  80  A9  40  C5  DF  F0  3E  B1
8020−  DE  0A  4A  85  DB  C5  DA  D0
8028−  26  E0  01  D0  11  20  8E  80
8030−  A5  DB  85  DA  09  80−91  DC
8038−  20  7E  80  20  7E  80  E8  20
```

```
8040- 8E 80 8A 91 DC 20 7E 80
8048- E0 FF D0 CA 4C 9E 80 91
8050- DC 20 7E 80 A5 DB 85 DA
8058- A2 01 4C 16 80 60 A9 FF
8060- C5 DE F0 03 E6 DE 60 A9
8068- 00 85 DE E6 DF 60 A9 00
8070- C5 DE F0 03 C6 DE 60 A9
8078- FF 85 DE C6 DF 60 A9 FF
8080- C5 DC F0 03 E6 DC 60 A9
8088- 00 85 DC E6 DD 60 A9 00
8090- C5 DC F0 03 C6 DC 60 A9
8098- FF 85 DC C6 DD 60 20 5E
80A0- 80 B1 DE 91 DC 20 6E 80
80AB- A2 01 20 8E 80 A9 FE 91
80B0- DC 20 7E 80 20 7E 80 4C
80B8- 16 80
```

运行下面的程序可处理压缩后的图形数据，并将图形显示出来:

```
5  REM PROGRAM 115.4
10  D$ = CHR$(4): HOME
20  PRINT D$;"BLOAD JB-2"
30  VTAB 10:HTAB 10: INPUT "PIC NAME:"; F$
40  PRINT D$;"BLOAD ";F$;"A$4000"
50  POKE 220,0:POKE 221, 64
60  HGR : POKE - 16302,0
70  CALL 32961
```

"JB-2"机器语言程序如下:

```
80C1- A9 20 85 DF A9 00 85
80C8- DE A0 00 4C 29 81 B1 DC
80D0- 85 DA 0A B0 0D A5 DA 91
80D8- DE 20 19 81 20 09 81 4C
80E0- CB 80 A5 DA 0A 4A 91 DE
80E8- 85 DA 20 19 81 B1 DC AA
80F0- CA 8A F0 0B 20 09 81 A5
80F8- DA 91 DE CA 4C F1 80 20
8100- 09 81 20 19 81 4C CB 80
8108- 60 A9 FF C5 LDE F0 03 E6
8110- DE 60 A9 00 85 DE E6 DF
8118- 60 A9 FF C5 DC F0 03 E6
8120- DC 60 A9 00 85 DC E6 DD
8128- 60 A9 40 C5 DF F0 D9 4C
8130- CE 80
```

116. 如何用键盘绘制低分辨率图形

运行下面的 BASIC 程序，并根据"菜单"提示进行操作，便可实现键盘控制的低分辨绘图，还可将图形存入磁盘或打印出来。

```
5  REM PROGRAM 116.1
10  REM PLOT
20  TEXT:HOME: I = 1
30  PRINT TAB(6)"* * * PROGRAM 10.2 * * *"
40  PRINT TAB(7)"KEYS:";: PRINT TAB(20)"MEANS": PRINT TAB(4)"------
```

```
————"
50  PRINT TAB (10)"U:" TAB(20)"UP"
60 PRINT TAB(10)"H:"TAB(20)"LEFT"
70  PRINT TAB (10)"J:" TAB(20)"RIGHT"
80  PRINT TAB (10)"N:" TAB (20)"DOWN"
90  PRINT TAB (10)"Y:" TAB (20)"UP- LEFT"
100  PRINT  TAB(10)"I:" TAB(20)"UP-RIGHT"
110  PRINT  TAB(10)"B:" TAB(20)"DOWN-LEFT"
120  PRINT  TAB(10)"M:" TAB(20)"DOWN-RIGHT"
130  PRINT  TAB(10)"S:" TAB(20)"SAVE-PLOT"
140  PRINT  TAB(10)"A:" TAB(20)"LOAD-PLOT      "
150  PRINT  TAB(10)"W:" TAB(20)"PRINT-PLOT"
160  PRINT  TAB(10)"T:" TAB(20)"LIST ORDERS"
170  PRINT  TAB(10)"Q:" TAB(20)"CLEAR SCREEN"
180  PRINT  TAB(10)"L:" TAB(20)"PLOT"
190  PRINT  TAB(10)"O:" TAB(20)"UNPLOT"
200  PRINT  TAB(10)"C:" TAB(20)"COLOR=(0——15)"
210  PRINT  TAB(10)"E:" TAB(20)"END"
220  PRINT : PRINT TAB(7) "YUN KEY:SPACE"
230  GET A $
240  IF A $ <  > " " THEN  340
250  HOME : VTAB 15: PRINT  TAB(5)
260  INPUT "WHAT IS X CO-ORD OF START:";X
270  PRINT TAB(5)
280  INPUT "WHAT IS Y CO-ORD OF START:";Y
290  IF X < 0 OR X > 39 OR Y < 0 OR Y > 39 THEN 250
300  HOME :C = 3 : GR
310  COLOR = C
320  PLOT X ,Y
330  GET A $
340  IF A $ = "U"THEN Y = Y - 1: GOTO 540
350  IF A $ = "I" THEN X = X + 1:Y = Y - 1: GOTO 540
360  IF A $ = "J" THEN X = X + 1: GOTO 540
370  IF A $ = "M" THEN X = X + 1:Y = Y + 1: GOTO 540
380  IF A $ = "N" THEN Y = Y + 1: GOTO 540
390  IF A $ = "B" THEN X = X - 1:Y = Y + 1: GOTO 540
400  IF A $ = "H" THEN X = X - 1: GOTO 540
410  IF A $ = "Y" THEN X = X - 1: Y = Y - 1: GOTO 540
420  IF A $ = "L" OR A $ = "O" THEN B $ = A $ : GOTO 580
```

```
430  IF A$ = "S" THEN 810
440  IF A$ = "A" THEN 890
450  IF A$ = "W" THEN 610
460  IF A$ = "C" THEN 510
470  IF A$ = "T" THEN HOME: TEXT : GOTO 20
480  IF A$ = "Q" THEN GR : TEXT: GOTO 250
490  IF A$ = "E"THEN TEXT : HOME: VTAB 15: PRINT TAB ( 17)"--END--": END
500  GOTO 330
510  HOME : VTAB 22 : HTAB 15
520  INPUT "COLOR=";C
530  COLOR= C: HOME
540  IF X < 0 THEN X = 0 : CALL - 198
550  IF X > 39 THEN X = 39: CALL - 198
560  IF Y > 39 THEN Y = 39: CALL - 198
570  IF Y < 0 THEN Y = 0 : CALL -198
580  IF B$ = "L" THEN PLOT X,Y: GOTO 330
590  PLOT X,Y: FOR P = 1 TO 20: NEXT P: COLOR= 0 : PLOT X,Y
600  COLOR= C: GOTO 330
610  REM COPY-GR
620  LOMEM: 16384
630  HGR
640  FOR I = 0 TO 39
650  FOR J = 0 TO 39
660 SC = SCRN( I,J)
670  IF SC > 0 THEN GOSUB 710
680  NEXT J,I
690  PR#1: PRINT CHR$ (17): PR#0
700  GOTO 20
710  IF SC = 4 OR SC = 12 THEN HC = 1: GOTO 760
720  IF SC = 3 OR SC = 1 THEN HC = 2: GOTO 760
730  IF SC = 8 OR SC = 9 OR SC = 11 OR SC = 13 THEN HC = 3: GOTO 760
740  IF SC = 6 OR SC = 7 OR SC = 2 THEN HC = 5 : GOTO 760
750 HC = 1
760  HCOLOR = HC
770  FOR K = 0 TO 3
780  HPLOT I * 7,J * 4+K TO (I+1) * 7-1,J * 4+K
790  NEXT K
800  RETURN
810  HOME: PRINT TAB( 10)"GIVE A NAME FOR THE IMAG"
```

```
820  INVERSE : PRINT "BE GOING TO STORE AFTER KEYING RETURN": NORMAL
830   INPUT "NAME:";N $ : HOME
840  D $  = CHR $ (4):F $  = "BSAVE" + N $  + ",A $ 400,L $ 400"
850   PRINT D $ : PRINT D $ ;F $
860   HOME : VTAB 23: PRINT TAB( 10)"STORE COMPLETEK"
870   FOR I = 1 TO 2000: NEXT I
880   GOTO 20
890   HOME : VTAB 23: PRINT TAB ( 10)"GIVE A NAME FOR THE IMAG"
900   PRINT TAB ( 10);" INPUT N $
910   GR: HOME
920  D $  = CHR $ (4): F $  = "BLOAD" + N $
930   PRINT D $ : PRINT D $ ;F $
940   GET A $
950   IF A $  = "W" THEN 610
960   GOTO 20
```

117. 如何用键盘绘制高分辨率图形

运行下面的 BASIC 程序，并根据“菜单”提示进行操作，便可实现键盘控制的高分辨率绘图，还可将图形存入磁盘或打印出来。

```
5   REM PROGRAM 117.1
10   REM HPLOT
20   HGR2 : HCOLOR = 3: HPLOT 0,0 TO 279,0TO 279,191 TO 0,191 TO 0,0
30   FOR I = 0 TO 279 STEP 10: HPLOT I,0 TO I,3: HPLOT I,187 TO I,191: NEXT I
40   FOR I = 0 TO 191 STEP 10: HPLOT 0, I TO 3,I: HPLOT 275,I TO 279,I: NEXT I
50   B $  = "L":C = 3
60   TEXT : HOME
70   ONERR GOTO 920
80   VTAB 10: HTAB 17: INPUT "X=";X
90   VTAB 12: HTAB 17: INPUT "Y=";Y
100   HOME : INVERSE
110   VTAB 10: HTAB 10: PRINT "<==MAKE AN IMAG==>"
120   FOR I = 1 TO 1500: NEXT I
130   HOME : NORMAL
140   GOTO 360
150   HTAB 10: PRINT "KEYS";: HTAB 20: PRINT ":";: PRINT "MEANS": HTAB 10:
PRINT "————————"
160   HTAB 10: PRINT "U";: HTAB 20: PRINT ":UP"
```

```
170  HTAB 10: PRINT "H";: HTAB 20: PRINT ":LEFT"
180  HTAB 10: PRINT "J";: HTAB 20: PRINT ":RIGHT"
190  HTAB 10: PRINT "N";: HTAB 20: PRINT ":DOWN"
200  HTAB 10: PRINT "Y";: HTAB 20: PRINT ":UP-LEFT"
210  HTAB 10: PRINT "I";: HTAB 20: PRINT ":UP-RIGHT"
220  HTAB 10: PRINT "B";: HTAB 20: PRINT ":DOWN-LEFT"
230  HTAB 10: PRINT "M";: HTAB 20: PRINT ":DOWN-RIGHT"
240  HTAB 10: PRINT "L";: HTAB 20: PRINT ": PLOT"
250  HTAB 10: PRINT "O";: HTAB 20: PRINT ":UNPLOT"
260  HTAB 10: PRINT "Q";: HTAB 20: PRINT ":CLERA SCREEN"
270  HTAB 10: PRINT "T";: HTAB 20: PRINT ":LIST ORDERS"
280  HTAB 10: PRINT "A";: HTAB 20: PRINT ":DISPLAYAN IMAG": HTAB 22: PRINT
" HAD BEEN MADE "
290  HTAB 10: PRINT "S";: HTAB 20: PRINT ":STORE ITINTO DISK"
300  HTAB 10: PRINT "W";: HTAB 20: PRINT ":ARD COPY ON PRINTER"
310  HTAB 10: PRINT "C";: HTAB 20: PRINT ":HCOLOR= (0--7)"
320  HTAB 10: PRINT "CTRL-E";: HTAB 20: PRINT ";GOTO END"
330  PRINT : HTAB 5: PRINT " RETURN,IF YOU KEY:SPACE-BAR"
340  GET E$ : IF E$ = "" THEN RETURN
350  GOTO 340
360  GOSUB 150
370  GOTO 570
380 W1 = PEEK (- 16384): IF W1 > 127 THEN A$ = CHR$ (W1 - 128): POKE -
16368,0: GOTO 420
390  HPLOT X,Y: HCOLOR= 0: HPLOT X,Y: HCOLOR= C
400  IF B$ = "L" THEN HPLOT X,Y
410  GOTO 380
420  IF A$ = "A" THEN 850
430  IF A$ = "U" OR A$ = "Y" OR A$ = "H" OR A$ = "J" OR A$ = "M" OR A$ = "N"
OR A$ = "B" OR A$ = "I" THEN GOSUB 770: GOTO 520
440  IF A$ = "Q" THEN 20
450  IF A$ = "T" THEN 560
460  IF A$ = "S" THEN GOSUB 960: GOTO 590
470  IF A$ = "W" THEN PR#0: GOTO 710
480  IF A$ = "" THEN TEXT: HOME: END
490  IF A$ = "L" OR A$ = "O" THEN B$ = A$ : GOTO 530
500  IF A$ = "C" THEN TEXT: HOME : VTAB 10: HTAB 15: INPUT "HCOLOR=";C:
POKE - 16299,0: POKE - 16304,0: HCOLOR= C: HPLOT X,Y: GOTO 380
510  CALL - 198: GOTO 380
```

```
520 C$ = A$
530  IF B$ = "L" THEN HPLOT X,Y: GOTO 380
540  HPLOT X,Y: FOR I = 1 TO 5: NEXT I: HCOLOR= 0: HPLOT X,Y
550  HCOLOR= 3: GOTO 380
560  TEXT : HOME : GOSUB 150
570  POKE - 16304,0: POKE - 16299,0: POKE - 16297,0
580  GOTO 380
590  HOME
600  TEXT : VTAB 10: HTAB 10: PRINT " GIVE A NAME FOR THE IMAG"
610  INVERSE : VTAB 20 : PRINT " BE GOING TO STORE AFTER KEYING RETURN"
620  INPUT N$ : FLASH
630  HOME
640  VTAB 15 : HTAB 20: PRINT "STORING"
650  NORMAL
660 F$ = "BSAVE" + N$ + ",A$4000,L$ 2000"
670  PRINT CHR$ (4);F$
680  VTAB 22: HTAB 10: PRINT "STORE COMPLETED"
690  FOR I = 1 TO 1000: NEXT I
700  GOTO 570
710  HOME: TEXT : HTAB 10: VTAB 10: FLASH : PRINT "HARD COPYING": NORMAL
720  PR#1: POKE 1145,75: POKE 1913,66: PRINT CHR$ (17)
730 PR#0
740  VTAB 20: PRINT "HARD COPY COMPLETED"
750  FOR I= 1 TO 1000: NEXT I
760  GOTO 570
770  IF A$ = "U" THEN Y = Y - 1: RETURN
780  IF A$ = "N" THEN Y = Y + 1: RETURN
790  IF A$ = "H" THEN X = X - 1: RETURN
800  IF A$ = "J" THEN X = X + 1: RETURN
810  IF A$ = "Y" THEN X = X - 1: Y = Y - 1: RETURN
820  IF A$ = "I" THEN X = X + 1: Y = Y - 1: RETURN
830  IF A$ = "B" THEN X = X - 1: Y = Y + 1: RETURN
840  IF A$ = "M" THEN X = X + 1: Y = Y + 1: RETURN
850  HOME: TEXT: VTAB 10: HTAB 10
860  PRINT "PLEASE TYPE IN THE IMAG'S NAME"
870  INVERSE : VTAB 20: PRINT " BE GOING TO DISPLAY AFTER KEYING RE-
TURN";: NORMAL
880  INPUT M$
890  POKE - 16304,0: POKE - 16299,0: POKE - 16297,0
```

```
900  PRINT CHR $ (4)"BLOAD";M $
910  GOTO 380
920  HOME : TEXT : HTAB 10: VTAB 10
930  PRINT "CAUTION! YOU HAVE AN ERR."
940  CALL - 198: GOTO 570
950  ONERR GOTO 920
960  HCOLOR= 0: HPLOT 0,0TO 297,0 TO 297,191 TO 0,191 TO 0,0
970  FOR I = 0 TO 279 STEP 10: HPLOT I,0 TO I,3: HPLOT I,187 TO I,191: NEXT I
980  FOR I= 0 TO 191 STEP 10: HPLOT 0, I TO 3,I: HPLOT 275,I TO 279, I: NEXT I
990  HCOLOR= 3: RETURN
```

118. 如何用汇编语言编写绘制高分辨率图形的程序

用 BASIC 语言编写绘制高分辨率图形的程序虽然简单、易读，但执行速度慢。为了提高绘图速度，可用汇编语言编写绘制高分辨率图形的程序。

一. 设定高分辨率图形显示方式

1. 设定高分辨率图形与文本混合第一页显示

使用 JSR ＄F3E2 命令，它相当于 BASIC 语言中的 HGR 命令。

2. 设定高分辨率图形全屏幕第二页显示

使用 JSR ＄F3D8 命令，它相当 BASIC 于语言中的 HGR2 命令。

3. 利用四对软开关设定屏幕显示方式

访问表 118–1 所示的内存单元，可将屏幕设定为某种显示状态。访问的方法是使用 LDA 或 STA 等命令。

表 118–1　中华学习机屏幕显示软开关

功　　能	地　址
选定图形显示方式	＄C051
选定文本显示方式	＄C051
选定全屏幕图形显示方示	＄C052
选定图形和文本混合显示方示	＄C053
选定第一页图形或文本显示方式	＄C054
选定第二页图形或文本显示方式	＄C055
选定低分辨率图形显示方式	＄C056
选定高分辨率图形显示方式	＄C057

4. 设定高分辨率绘图页

往 ＄E6 内存单元送相应的数，可设定在高分辨率第 n 页图。设定高分辨率绘图页后，使用 JSR ＄F3F2 可将该页内存单元清干净。往 ＄E6 内存单元送的数与设定的绘图

页的对应关系如下表 118–2 表所示。

表 118–2 中华学习机高分辨率绘图页的设定

功　　能	往 $ E6 内存单元送的数
设定高分辨率第一页绘图	$ 20
设定高分辨率第二页绘图	$ 40
设定高分辨率第三页绘图	$ 60
设定高分辨率第四页绘图	$ 80
设定高分辨率第五页绘图	$ A0

二. 设定图形颜色

往 $ E4 内存单元送相应的数，可设定高分辨率图形颜色。往 $ E4 内存单元送的数与设定颜色的对应关系如 118–3 表所示。

表 118–3 往 $ E4 内存单元送的数与设定颜色的对应关系

数值	颜色	数值	颜色
$ 00	黑色 1	$ 80	黑色 2
$ 2A	绿色	$ AA	橙色
$ 55	紫色	$ D5	蓝色
$ 7F	白色 1	$ FF	白色 2

三. 绘图命令

1. 绘点

将该点的水平坐标值的低位送入变址寄存器 X，高位送入寄存器Y，将该点的垂直坐标值送入累加器 A，然后调用起始地址为 $ F457 的绘点子程序。

2. 绘线

先绘起点，然后将终止的水平坐标值的低位送入累加器 A，高位送入变址寄存器 X，将终点垂直坐标值送入变址寄存器 Y 。最后调用起始地址为 $ F53A 的绘线子程序。

3. 将整个屏幕的颜色与刚绘过的点或线的颜色一样。

下面用汇编语言绘一长方形，并在长方形内绘一点：

```
0300- 20 E2 F3    JSR  $F3E2       0322- 20 3A F5    JSR  $F53A
0303- 8D 52 C0    STA  $C052       0325- A9 0A       LDA  #$0A
0306- A9 FF       LDA  #$FF        0327- A2 00       LDX  #$00
0308- 85 E4       STA  $E4         0329- A0 A0       LDY  #$A0
030A- A9 0A       LDA  #$0A        032B- 20 3A F5    JSR  $F53A
030C- A2 0A       LDX  #$0A        032E- A9 0A       LDA  #$0A
```

```
030E- A0 00      LDY  #$00
0310- 20 57 F4   JSR  $F457
0313- A9 0A      LDA  #$0A
0315- A2 01      LDX  #$01
0317- A0 0A      LDY  #$0A
0319- 20 3A F5   JSR  $F53A
031C- A9 0A      LDA  #$0A
031E- A2 01      LDX  #$01
0320- A0 A0      LDY  #$A0
0330- A2 00      LDX  #$00
0332- A0 0A      LDY  #$0A
0334- 20 3A F5   JSR  $F53A
0337- A9 80      LDA  #$80
0339- A2 50      LDX  #$50
033B- A0 00      LDY  #$00
033D- 20 57 F4   JSR  #$F457
0340- 60         RTS
```

119. 如何用汇编语言编写绘制低分辨率图形的程序

为了提高绘制低分辨率图形的速度，可用汇编语言编写绘制低分辨率图形的程序。

一. 设定低分辨率图形显示方示

使用 JSR FB40 命令可设定屏幕为低分辨率图形与文本混合第一页显示方式，相当于执行 BASIC 语言中的 GR 命令。

如要设定其它低分辨率图形显示方式，可参看“汇编语言编写绘制高分辨图形的程序”一文中有关屏幕显示软开关的内容。

如果要清文本屏幕，可使用 JSR ＄FC58 命令，它相当于 BASIC 语言中的 HOME 命令。

二. 设定图形颜色

将彩色代码数据存入 ＄30 内存单元或累加器 A，然后调用起始地址为 ＄F864 的子程序。彩色代码数据与颜色的对应关系如下表 119-1 所示。

表 119-1　颜色代码数据与颜色的对应关系

送入＄30 内存单元的数据	送入累加器 A 的数据	颜 色
＄00	＄00	黑色
＄11	＄01	深红
＄22	＄02	深蓝色
＄33	＄03	紫色
＄44	＄04	深绿色
＄55	＄05	灰色 1
＄66	＄06	蓝色
＄77	＄07	浅蓝色
＄88	＄08	棕色
＄99	＄09	橙色
＄AA	＄0A	灰色 2
＄BB	＄0B	粉红色
＄CC	＄0C	浅绿
＄DD	＄0D	黄色
＄EE	＄0E	海蓝色
＄FF	＄0F	白色

三. 绘图命令

1. 绘点

将色点(色块)的水平坐标值送入变址寄存器 Y,垂直坐标值送入累加器 A，然后调用起始地址为 $ F800 的子程序。

如果所绘的点与刚绘完的点在同一行上，即累加器 A 内值不变，变址寄存器 Y 内的值改变后，调用起始地址为 $ F80E 的子程序来绘点。这样可使绘图速度加快。

2. 绘水平线

将水平线的垂直坐标值送入累加器 A，将水平线左端点的水平坐标值送入变址寄存器 Y，右端点的水平坐标值送入 $ 2C 内存单元，然后调用起始地址为 $ F819 的子程序。

3. 绘垂直线

将垂直线的水平坐标值送入变址寄存器 Y，将垂直线上端点的垂直坐标值送入累加器 A，下端点的垂直坐标值送入 $ 2D 内存单元，然后调用起始地址为 $ F828 的子程序。

4. 使低分辨率显示屏幕为黑色

调用起始地址为 $ F832 的子程序，可使整个低分辨率图形屏幕 (48 行)绘为黑色。

调用起始地址为 $ F836 的子程序，可使低分辨率图形屏幕上 40 行绘为黑色。

四. SCRN 函数

将某色块水平坐标值送入变址寄存器 Y，垂直坐标值送入累加器 A，然后调用起始地址为 $ F871 的子程序，可将该色块的颜色代码值送入累加器 A 。

下面用汇编语言绘一色块和一彩色对角线。

绘一个色块的程序

```
0300- 20 58 FC        JSR   $FC58
0303- 20 40 FB        JSR   $FB40
0306- A9 99           LDA   #$99
0308- 85 30           STA   $30
030A- 20 64 F8        JSR   $F864
030D- A9 0A           LDA   #$0A
030F- A0 0A           LDY   #$0A
0311- 20 00 F8        JSR   $F800
0314- 60              RTS
```

绘一彩色对角线的程序

```
0300- 20 58 FC        JSR   $FC58
0303- 20 40 FB        JSR   $FB40
0306- A9 0B           LDA   #$0B
```

```
0308-  20 64 F8        JSR   $F864
030B-  A0 00           LDY   #$00
030D-  98              TYA
030E-  20 00 F8        JSR   $F800
0311-  C8              INY
0312-  C0 28           CPY   #$28
0314-  D0 F7           BNE   $030D
0316-  60              RTS
```

120. 如何用 PRINT 语句在高分辨率画面上写字符

在高分辨率画面上印出字符的方法很多，这里介绍一种用 PRINT 语句在高分辨率画面上写字符的方法。这种方法使用方便，就象在文本屏幕上写字符一样。

每个字符由 5＊7 的点阵组成，占用八个内存单元，保证字符间有一行或列色点的间距。所有字符数据(每个字符八个数据)组合在一起形成一个表，存放在 $6000—$61FF 内存单元中。字符数据的确定方法可参看问题“如何使用 POKE 语句绘高分辨率图形”一文。

另外，在 $300—$372 内存单元中存放一个查表机器语言子程序。使用时，往 $36、$37(即 54、55)内存单元中分别送入 $00、$03 即可。不能使用 CALL 768(或＊300G)命令。

```
6000-  00 1C 22 2A 3A 1A 02 3C      6050-  00 20 20 20 20 20 22 1C
6008-  00 08 14 22 22 3E 22 22      6058-  00 22 12 0A 06 0A 12 22
6010-  00 1E 22 22 1E 22 22 1E      6060-  00 02 02 02 02 02 02 3E
6018-  00 1C 22 02 02 02 22 1C      6068-  00 22 36 2A 2A 22 22 22
6020-  00 1E 22 22 22 22 22 1E      6070-  00 22 22 26 2A 32 22 22
6028-  00 3E 02 02 1E 02 02 3E      6078-  00 1C 22 22 22 22 22 1C
6030-  00 3E 02 02 1E 02 02 02      6080-  00 1E 22 22 1E 02 02 02
6038-  00 3C 02 02 02 32 22 3C      6088-  00 1C 22 22 22 2A 12 2C
6040-  00 22 22 22 3E 22 22 22      6090-  00 1E 22 22 1E 0A 12 22
6048-  00 1C 08 08 08 08 08 1C      6098-  00 1C 22 02 1C 20 22 1C

60A0-  00 3E 08 08 08 08 08 08      60D0-  00 3E 20 10 08 04 02 3E
60A8-  00 22 22 22 22 22 22 1C      60D8-  00 3E 06 06 06 06 06 3E
60B0-  00 22 22 22 22 22 14 08      60E0-  00 00 02 04 0A 12 20 00
60B8-  00 22 22 22 2A 2A 36 22      60E8-  00 3E 30 30 30 30 30 3E
60C0-  00 22 22 14 08 14 22 22      60F0-  00 00 00 08 14 22 00 00
60C8-  00 22 22 14 08 08 08 08      60F8-  00 00 00 00 00 00 00 3E
```

```
6100- 00 00 00 00 00 00 00 00
6108- 00 08 08 08 08 08 00 08
6110- 00 14 14 14 00 00 00 00
6118- 00 14 14 3E 14 3E 14 14
6120- 00 08 3C 0A 1C 28 1E 08
6128- 00 06 26 10 08 04 32 30
6130- 00 04 0A 0A 04 2A 12 2C
6138- 00 08 08 08 00 00 00 00
```

```
6140- 00 08 04 02 02 02 04 08
6148- 00 08 10 20 20 20 10 08
6150- 00 08 2A 1C 08 1C 2A 08
6158- 00 00 08 08 3E 08 08 00
6160- 00 00 00 00 00 08 08 04
6168- 00 00 00 00 3E 00 00 00
6170- 00 00 00 00 00 00 00 08
6178- 00 00 20 10 08 04 02 00
6180- 00 1C 22 32 2A 26 22 1C
6188- 00 08 0C 08 08 08 08 1C
6190- 00 1C 22 20 18 04 02 3E
6198- 00 3E 20 10 18 20 22 1C
61A0- 00 10 18 14 12 3E 10 10
61A8- 00 3E 02 1E 20 20 22 1C
61B0- 00 38 04 02 1E 22 22 1C
61B8- 00 3E 20 10 08 04 04 04
61C0- 00 1C 22 22 1C 22 22 1C
61C8- 00 1C 22 22 3C 20 10 0E
61D0- 00 00 00 08 00 08 00 00
61D8- 00 00 00 08 08 00 08 04
61E0- 00 10 08 04 02 04 08 10
61E8- 00 00 00 3E 00 3E 00 00
61F0- 00 04 08 10 20 10 08 04
61F8- 00 1C 22 10 08 08 00 08
```

```
0300- 48 8D 50 C0 8D 52 C0 8D
0308- 57 C0 C9 8D F0 3B 29 3F
0310- 0A 0A 0A 85 0E A9 60 69
0318- 00 85 0F A5 25 0A 0A 0A
0320- 85 00 8A 48 98 48 A2 08
0328- A0 00 20 4D 03 18 A5 04
0330- 65 24 85 04 A5 05 69 00
0338- 85 05 B1 0E 91 04 E6 00
0340- E6 0E CA D0 E5 68 A8 68
0348- AA 68 4C F0 FD A5 00 0A
0350- 0A 29 1C 85 05 A5 00 6A
0358- 6A 6A 6A 29 03 05 05 09
0360- 20 85 05 05 00 6A 29 E0
0368- 85 04 6A 6A 29 18 05 04
0370- 85 04 60
```

举例如下:

```
3  REM PROGRAM 120.1
5  D$ = CHR$(4): PRINT D$;"BLOAD H-P-1"; PRINT D$;"BLOADH-P-2"
10 HGR: POKE - 16302,0
15 POKE 54,0: POKE 55, 3
20 FOR I = 1 TO 7
25 VTAB 12: HTAB 7: PRINT "* * * * HPLOT-PRINT TEXT * * * *"
30 FOR J = 1 TO 1000: NEXT J
35 HCOLOR = I: HPLOT 1, 1: CALL 62454
40 NEXT I
```

```
45  HCOLOR = 5: HPLOT 1,1: CALL 62454
50  A $ = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
55  A $ = MID $ (A $ ,2,25) + LEFT $ ( A $ ,1 )
60  VTAB 12:HTAB 7: PRINT A $
65  FOR J = 1 TO 50 : NEXT J
70  GOTO 55
```

121. 如何使两页图形合并

在高分辨率第一页和第二页分别建立两个图形，然后将它们合并为一个，并在高分辨率第一页显示出来。

一.BASIC 程序法

在磁盘中存有要合并的两个图形数据，然后运行下面的程序，便可完成合并工作。

```
5   REM PROGRAM 121.1
10  HOME : VTAB 10: HTAB 10
20  INPIUT "PIC-1 NAME:";F1 $
30  VTAB 12: HTAB 10: INPUT"PIC-2 NAME:";F2 $
40  D $ = CHR $ (4)
50  PRINT D $ ; "BLOAD ";F1 $ ;",A $ 2000
60  PRINT D $ ;"BLOAD";F2 $ ;",A $ 4000
70  POKE - 16304,0: POKE - 16297,0
80  FOR I = 1 TO 10 AT EP3: READ S: POKE 768 + I, S: NEXT I:DATA 173,13,141,96
90  FOR I = 8192 TO 16383
100  N% = I / 256: M = I - N% * 256: POKE 769, M: POKE 770,N%
110  POKE 772,M: POKE 773,N% + 32: POKE 775,M: POKE 776,N%
120  CALL 768
130  NEXT I
```

二.机器语言程序法

运行下面 BASIC 程序，用它调用图形合并机器语言程序，机器语言程序存放在以 $ 300 为起始地址的内存区域中。

BASIC 程序如下:

```
5   REM PROGRAM 121.2
10  HOME: VTAB 10: HTAB 10
20  INPUT "PIC-1 NAME:";F1 $
30  VTAB 12: HTAB 10: INPUT "PIC-2 NAME:";F2 $
40  D $ = CHR $ (4)
50  PRINT D $ ;"BLOAD";F1 $ ;",A $ 2000"
60  PRINT D $ ;"BLOAD";F2 $ ;",A $ 4000"
70  CALL 768
```

机器语言程序如下:

```
0300- A9 FF 85 06 85 08 A9 F1
0308- 85 07 A9 3F 85 09 E6 06
0310- D0 02 E6 07 E6 08 D0 02
0318- E6 09 A2 00 A1 06 01 08
0320- 81 06 A9 3F C5 07 D0 E6
0328- A9 FF C5 06 D0 E0 8D 50
0330- C0 8D 52 C0 8D 54 C0 8D
0338- 57 C0 60
```

122. 如何将高分辨率图形放大或缩小

运行下面的 BASIC 程序后，键入&D X0，Y0 命令，即可将高分辨率第二页图形的某一局部图形[该局部图形的左上角顶点坐标为(X0，Y0)，其中 0<=X<=20，0<=Y<=96]放大到高分辨率第一页上；键入&X X0，Y0 命令，即可将高分辨率第二页图形缩小到高分辨率第一页某一块 140*96 点阵的图形上[(X0，Y0)为该块图形左上负顶点坐标。]。

BASIC 程序如下:

```
5  REM PROGRAM 123.1
10  D$ = CHR$ (4)
20  PRINT D$;"BLOAD PIC-1"
30  PRINT D$;"BLOAD PIC-2"
40  PRINT D$;"BLOAD PIC-3"
50  POKE 1014,0: POKE 1015,3
60  HOME : VTAB 10: HTAB 10 : INPUT "PIC NAME:";F$
70  PRINT D$;"BLOAD";F$;"A$4000
```

BASIC 程序调用的“PIC-1”机器语言用来处理&命令和调用放大或缩小的机器语言子程序“PIC-2”与“PIC-3”。各机器语言程序如下:

```
PIC-1
0300- A0 00 B1 B8 48 C8 20 98
0308- D9 68 C9 44 D0 06 20 2D
0310- 03 4C 00 60 C9 58 D0 29
0318- 20 2D 03 2C 50 C0 2C 55
0320- C0 2C 57 C0 2C 54 C0 2C
0328- 52 C0 4C 00 61 20 46 E7
0330- 86 01 A5 50 85 00 C9 15
0338- B0 07 A5 01 C9 60 B0 01
0340- 60
PIC-2
6000- A9 00 85 05 20 E2 F3 2C
6008- 52 C0 A9 00 85 08 85 09
6010- A9 20 85 E6 A5 05 20 11
6018- F4 A5 26 85 06 A5 27 85
6020- 07 E6 05 A5 05 20 11 F4
6028- A5 26 85 0A A5 27 85 0B
6030- A9 40 85 E6 A5 01 20 11
6038- F4 18 A5 26 65 00 85 26
6040- A4 09 B1 26 85 04 A2 00
6048- A0 04 A5 04 4A 76 02 46
6050- 04 76 02 88 D0 F4 E8 E0
6058- 02 D0 ED 06 02 26 03 66
```

```
6060- 02 A4 08 A5 02 91 06 91
6068- 0A C8 A5 03 91 06 91 0A
670- E6 09 E6 08 E6 08 A5 08
6078- C9 28 D0 C4 E6 01 E6 05
6080- A5 05 C9 1 C0 D0 84 60
```

PIC-3

```
6100- A9 00 85 08 A9 00 85 0C
6108- A9 40 85 E6 A5 08 20 11
6110- F4 A5 26 85 06 A5 27 85
6118- 07 E6 08 A5 08 20 11 F4
6120- A5 26 85 0A A5 27 85 0B
6128- A9 20 85 E6 A5 01 20 11
6130- F4 A5 00 85 0D A4 0C B1
6138- 06 11 0A 85 02 C8 B1 06
6140- 11 0A 85 03 E6 0C E6 0C
6148- A0 03 46 02 66 04 46 02
6150- 66 05 88 D0 F5 46 02 66
6158- 04 46 03 66 05 A0 03 46
6160- 03 66 04 46 03 66 05 88
6168- D0 F5 A5 04 05 05 18 6A
6170- A4 0D 91 26 E6 0D A4 0C
6178- C0 28 D0 BB E6 08 E6 01
6180- A5 08 C9 C0 F0 03 4C 04
6188- 61 60
```

123. 如何用磁带机存取 BASIC 程序

中华学习机可以用磁带机（即家用录音机）将主机内存中的 BASIC 程序存入磁带中，也可以从磁带中将程序输入到主机内存中。

1、如何用磁带机将 BASIC 程序存入磁带中

(1) 将磁带倒至起始处，并将计数器置零。

(2) 通过计算机键盘键入:

]SAVE″文件名″

但不按回车键同时按下磁带机的″PLAY″和″RECORD″键

(3) 按下回车键。当计算机响完第二声″嘟″时，表示程序已经装完，屏幕显示提示符″]″。

(4) 按磁带机的 STOP 键，录制完毕。

如果不键入“文件名”也可将文件存入磁带，但在取出程序时需借助计数器找到文件在磁带上的位置。如果在存入下一个程序，应将磁带转至第一个程序结尾处，再按上述步骤进行。

2、如何用磁带机将磁带中的 BASIC 程序取出

(1) 通过键盘键入:

]LOAD “文件名”

但不按回车键。

(2) 装入存放程序的磁带，按下磁带机的 PLAY 键，再按回车键。

(3) 当计算机响一声″嘟″后，程序已全部装入主机，这时屏幕会显示出被装入的文件名及 BASIC 提示符″]″。当装入的文件不是所需要的文件时，屏幕会提示已装入的文件名和以下字样信息:

HAS BEENLOADED

并继续装入后面的文件，直到装入 LOAD 语句中所指定的文件止。

(4) 按 STOP 键，使录音机停止转动。

如果没给文件名，则只将第一个文件调入内存，然后显示"]"提示符。

在使用磁带机存取程序时，如果失败，可调整磁带机音量（由小向大调整），重新操作一遍。还应注意经常清洗磁头。

124. 如何进行高分辨率页局部清屏

有时需将高分辨率画页某一窗口的内容清除，而 HGR,HGR2 只能将第一、二页画面全部清掉，下面的机器语言程序提供了两种形式的局部清屏：一般局部清屏与有趣味拉幕式局部清屏。

1、确定需处理的画页。$ E6 单元放入 $ 20 为高分辨率第一页；$ 40 为第二页。

2、确定需清除画面的范围。$ 00,$ 01 单元放入需清除窗口的左上角的坐标 X1,Y1；$ 02,$ 03 单元放入右下角的坐标 X2,Y2(其中：$ 00< = X< = $ 28 $ 00< = Y< = $ C0)。

3、调用程序：600G(CALL 24576)为一般局部清屏；6017G(CALL 24599)为拉幕式清屏。

```
6000-  20 6D 60 A6 01 20 60 60
6008- A9 00 91 06 C8 C4 02 D0
6010- F9 E8 E4 03 D0 EF 60 20
6018- 6D 60 A5 01 85 0A A6 01
6020- E8 20 60 60 CA BD 00 94
6028- 85 08 BD 00 95 85 09 B1
6030- 06 91 08 C8 04 02 D0 F7
6040- C0 CA A4 00 BD 00 94 85
6048- 08 BD 00 95 85 09 A9 00
6050- 91 08 C8 C4 02 D0 F9 E6
6058- 0A A5 03 C5 0A D0 BF 60
6060- A4 00 BD 00 94 85 06 BD
6068- 00 95 85 07 60 A2 00 BA
6070- 48 20 44 F4 68 AA A5 26
6078- 9D 00 94 A5 27 9D 00 95
6080- E8 E0 C0 D0 EA 60 00 00
```

五、磁盘操作系统

125. 什么是DOS

由于我们使用的微型计算机内存有限，并且断电后内存中的信息消失，不能将程序或数据长期地永久地保存，为此我们可以把这些程序和数据存放在计算机的外存储器中。为了协调计算机内外存储器之间信息的传递，计算机中必须有一个管理程序，这个管理程序叫磁盘操作系统（DISK OPERATION SYSTEM），简称DOS。

不同型号的微型机其DOS也不相同。对同种型号的计算机，DOS也有不同的版本，中华习机有以下不同的DOS版本：

DOS 3.0　　1978年6月研制
DOS 3.1　　1978年7月研制
DOS 3.2　　1979年2月研制
DOS 3.2.1　1979年8月研制
DOS 3.3　　1980年8月研制
ProDOS　　1983年研制

目前中华学习机上以DOS 3.3使用的最广泛。

126. 如何对磁盘进行初始化

厂家生产的磁盘适应各种型号的计算机用，一张刚买来的新磁盘不能立即用作存取盘，要先得在用户的计算机上对这张盘进行初始化（INITIALINE）工作，也称格式化。

中华学习机磁盘初始化的步骤如下：

（1）　将DOS 3.3系统主盘插入1号驱动器，然后打开主机开关（也可在开机情况下，键入PR#6热启动DOS 3.3系统）。

（2）　当出现提示符"]"后，根据需要键入一个问候程序，下面就是一个问候程序的例子：

```
 5 REM PROGRAM 126.1
10 HOME
20 PRINT "APPLE ][ DOS DISK PROGRAM 89.9.8"
30 PRINT CHR $ (4);"CATALOG"
40 END
```

（3）　拿出DOS 3.3系统盘，放入新磁盘，键入命令：INIT HELLO。其中HELLO是问候程序的文件名。这时驱动器工作的红灯亮并发出转动声，约一分钟后驱动器红灯灭，格式化完成。

一张在中华学习机上初始化后的磁盘，就可以存储DOS文件了。

127. 什么是文件、文件有哪些类型

文件就是一些相关信息的集合，它们存储在计算机的外部存储设备中，如磁盘等。文件可以表现为一段文字，也可以表现为一段程序或一批数据，每个文件都有一个特有名字，称为该文件的文件名。在磁盘操作系统下，用 DOS 命令对文件进行存取或管理。文件名的使用规定如下：

(1) 文件名的长度必须在 1 到 30 个字符之间，超出的字符无效而被舍掉；

(2) 文件名的第一个字符必须是英文大写字母；

(3) 任何键盘输入的字符都可以用作文件名的一部分，但是逗号“,”不能用。

文件分为五种类型，如下：

表 127–1

代码	意义
A	APPLESOFT 程序
B	机器语言文件
I	整数 BASIC 程序
T	文本文件
R	可被执行的二进制文件

128. DOS 的基本命令有哪些

DOS 命令的一般格式是：

DOS 命令　f [, Ss] [, Vv] [, Dd]

其中参数 f 是文件名；参数 s 是指驱动器接口卡所在的插槽号，一般驱动器卡插于 6 号槽，即 S6；参数 v 是指磁盘的卷号，它的范围是 1 到 254 之间的整数，每张磁盘的卷号在使用 INIT 命令中就应指定了，若 INIT 命令没有指定卷号，则卷号为 254；参数 d 是指磁盘所在驱动器的号。以上 DOS 命令格式中“[ ]”的可选项可以以任意次序排列，如果省略则以当前所使用的内容为准。

一. CATALOG 命令

这是列磁盘中文件目录的命令，一般格式是：

CATALOG [, Ss] [, Dd] [, Vv]

如：CATALOG, D2 命令为指定显示 D2 驱动器中的文件目录。

二. LOAD 命令

这是装入命令，将从磁盘中读出的程序文件装入到主机内存，一般格式为：

LOAD f [, Ss] [, Dd] [, Vv]

三. RUN 命令

这是运行命令，一般格式是：

RUN f [, Ss] [, Vv] [, Dd]

RUN 命令的作用是装入并且运行磁盘中的某个程序。

四. SAVE 命令

这是存入命令，一般格式是:

SAVE f [, Ss] [, Vv] [, Dd]

这个命令是把主机内的程序保存到磁盘中，如: SAVE ABC, D2 命令，是把内存中的程序以文件名 ABC 存入 2 号磁盘。SAVE 命令结束后，可以用 CATALOG 命令来检查程序是否已经复制完成。

五. DELETE 命令

这是删除命令，它将磁盘上指定的文件删除。它的一般格式是:

DELETE f [, Ss] [, Dd] [, Vv]

六. LOCK 命令

这是加锁命令，一般格式是:

LOCK f [, Ss] [, Dd] [, Vv]

这个命令是为磁盘上指定的文件加锁。文件加锁后就不能被删除或重写了。加锁的文件在磁盘目录栏中以“*”标记。

七. UNLOCK 命令

这是去锁命令，一般格式是:

UNLOCK f [, Ss] [, Dd] [, Vv]

八. RENAME 命令

这是更改磁盘文件名的命令，一般格式是:

RENAME f1, f2 [, Dd] [, Ss] [, Vv]

其中 f1 代表文件的旧名，f2 代表文件的新名。使用这个命令时要避免新的文件名与磁盘中的其它文件名重复。

九. VERIFY 命令

这是检验磁盘中的文件是否完整的命令，一般格式是:

VERIFY f [, Dd] [, Ss] [, Vv]

这个命令可以检查文件存到磁盘上的过程是否正确，所占的扇区是否有物理损坏。DOS 3.3 版本在存储文件完后都自动执行 VERIFY 操作。若 VERIFY 操作后显示“I/O ERROR”，说明存储文件出错，应该重新存盘。

129. 什么叫顺序文件、什么叫随机文件

中华学习机操作系统 DOS 可以提供两种类型的磁盘文件，即顺序文件和随机文件。顺序文件是在存储器中按线性顺序排列，并以其物理顺序存取，也可称顺序存取文件。

随机文件在存取时可不考虑记录存放的排列次序，任意进行某个记录的存取。随机文件也称随机存取文件。

130. 如何写入和读出顺序文件

建立一个顺序文件必须有下面四个步骤:

(1) 打开文件 OPEN 命令

(2) 说明写文件操作 WRITE 命令

(3) 写入数据 PRINT 命令

(4) 关闭文件 CLOSE 命令

例如，下面就是一个建立顺序文件的程序:

```
5  REM PROGRAM 130.1
10 D$=CHR$(4)
20 PRINT D$;"OPEN TET"
30 PRINT D$;"WRITE TET"
40 PRINT "TOKYO"
50 PRINT "NAGOYA"
60 PRINT "OOSAKA"
70 PRINT D$;"CLOSE TET"
80 END
```

此程序运行后，将在磁盘上建立了一个 T 型文件 TET，它的内容如下:

TOKYO ↓ NAGOYA ↓ OOSAKA ↓

检索一个磁盘中的顺序文件必须有下面四个步骤:

(1) 打开文件 OPEN 命令

(2) 说明读文件操作 READ 命令

(3) 读出数据 INPUT 语句

(4) 关闭文件 CLOSE 命令

例如，要读出上面建立的 TET 文件，可以运行下程序:

```
5  REM PROGRAM 130.2
10 D$=CHR$(4)
20 PRINT D$;"OPEN TET"
30 PRINT D$;"READ TET"
40 INPUT A$
50 INPUT B$
60 INPUT C$
70 PRINT D$;"CLOSE TET"$
80 PRINT A$,B$,C$
90 END
```

131. 如何写入和读出随机文件

随机文件的写出与读出和顺序文件大致相同。

(1) 打开随机文件命令:

OPEN f, Li

其中参数 i 表示记录的长度，取值在 0–32767 之间，每次打开同一个文件时，所用长度必须一致，这一要求是根据随机文件的特点确定的。长度的计算应包括每个记录字符总数和回车符↓（占一个字节），i 要以文件中最大的记录长度来设定。

(2) 关闭随机文件的命令

CLOSE f

这个命令的作用同顺序文件中的关闭命令是一致的。

(3) 写随机文件命令

WRITE f, R r

其中参数 r 表示写入的数据所在的记录号。如

60 PRINT D$; "WRITE TB, R4"

这个语句表示后面的 PRINT 语句中的内容将写在文件 TB 的第 4 号记录中。

例如：向文件名为 PAC 的随机文件的第 10 号记录写入数据“BASIC”，向第 12 号记写入数据“COBOL”；设文件记录长度为 15 个字节。建立这个文件的程序如下:

```
5  REM  PROGRAM 131.1
10  D$ =CHR$(4)
20  PRINT  D$; "OPEN PAC,L15"
30  PRINT  D$; "WRITE PAC,R10"
40  PRINT  "BASIC"
50  PRINT  D$; "WRITE PAC,R12"
60  PRINT  "COBOL"
70  PRINT  D$; "CLOSE PAC"
80  END
```

(4) 读随机文件命令

READ f, Rr

这个语句表示读出文件名为 f 的随机文件的第 r 号记录。例如检索上面建立的 PAC 文件中第 12 号记号可以用下面的程序:

```
5  REM PROGRAM 131.2
10  D$ =CHR$(4)
20  PRINT  D$; "OPEN PAC,L15"
30  PRINT  D$; "READ PAC,R12"
40  INPUT  X$
50  PRINT D$; "CLOSE PAC"
60  PRINT  X$
70  END
```

132. 在随机文件中怎样使用 POSITION 命令

中华学习机上使用的 DOS 3 · 3 操作系统，顺序文件命令之一的 POSITION 命令，是一个移动定位指标的命令，它可以将文件中记录的指针移至当前位置往下若干记录的起始位置，例如，假设有一个打开的 100 条记录的顺序文件 MX，用命令：POSITION MX，R50 即将指针移至第 51 个记录上，下面若紧接着用命令：READ MX 就将从第 51 个记录读起。这是在顺序文本文件中 POSITION 命令的用法，一般来说在随机文本文件中没有用到 POSITION 命令，但是根据随机文本文件与顺序文件的结构特点，可以把随机文件中一个记录当成一个顺序文件，当读写这个随机文件的一个记录时，可以用 POSITION 命令读写这个记录中的任一项。为了更清楚地说明上面的原理，下面举一个例子：

首先建立一个随机文件，它有 3 个记录，每一个记录设有 10 项内容：

```
5   REM PROGRAM 132.1
10  D$ =CHR$(4)
20  PRINT D$;"OPEN NX, L100"
30  FOR I=1 TO 3
40  PRINT D$;"WRITE NX, R" I
50  FOR J=1 TO 10
60  PRINT I;  "·";  J
70  NEXT J
80  NEXT I
90  PRINT D$;"CLOSE NX"
```

下面我们要检索出每个记录第 5 项的内容：

```
5   REM PROGRAM 132.2
10  D$ =CHR$(4)
20  PRINT D$;"OPEN NX, L50"
30  FOR I=1 TO 3
40  PRINT D$;"READ NX,  R";  I
50  PRINT D$;"POSITION NX, R4 "
60  PRINT D$;"READ NX"
70  INPUT A$
80  PRINT A
90  NEXT I
100  PRINT D$;"CLOSE NX"
```

这样，我们不但可以方便地读写随机文件中的任一记录，而且可以读写文件任一记录的任一项。

133. 如何补写文件的内容

建立顺序文件时用 OPEN 命令打开文件，此时文件的读／写定位指针将设在文件的第一字符上（即零字节）。在顺序文件中用 APPEND 命令可以增补文件，APPEND 命令使用后会将文件读／写定位指针设到文件最后一个字节上，这时就可以向该文件中增补数据了。在随机文件的补写数据时，只需指定记录号即可。

例如通过以下程序建立了一个文件，名称为 EX201（间接名称为 F $）：

```
5   REM PROGRAM 133.1
10  D $ =CHR $ (4)
20  INPUT "FILE NAME="; F $
30  PRINT D $ ;"OPEN "; F $
40  PRINT D $ ;"DELETE "; F $
50  PRINT D $ ;"OPEN ";F $
60  PRINT D $ ;"WRITE "; F $
70  FOR I = 1 TO 10
80  PRINT I
90  NEXT I
100 PRINT D $ ;"CLOSE ";F $
110 END
```

可通过以下程序在已建立的原数据文件之后补写数据：

```
5   REM PROGRAM 133.2
10  D $ =CHR $ (4)
20  INPUT "FILE NAME= ";F $
30  PRINT D $ ;"APPEND ";F $
40  PRINT D $ ;"WRITE ";F $
50  FOR I = 1 TO 10
60  PRINT I * I
70  NEXT I
80  PRINT D $ ;"CLOSE ";F $
90  END
```

通过以下程序，可从数据文件 EX201 中取出以上两个程序所存入的原存数据和补写数据。

```
5   REM PROGRAM 133.3
10  D $ =CHR $ (4)
20  DIM A(20)
30  INPUT "FILE NAME=";C $
40  PRINT  D $ ;"OPEN ";C $
50  PRINT  D $ ;"READ ";C $
```

```
60   FOR I = 1 TO 20
70   INPUT A(I)
80   NEXT I
90   PRINT D$;"CLOSE";C$
100  FOR I = 1 TO 20
110  PRINT A(I),
120  NEXT I
130  END
```

134. 如何监视文件的写入和读出

通常在磁盘上写入文件和读出文件时，屏幕上没有什么显示。为了能从屏幕上监视程序的运行情况，可以用磁盘监视命令 MON。它的格式如下：

MON C, I, O

其中 C 表示对磁盘操作命令（如 OPEN、READ 等）的监视；I 表示对由磁盘输入的监视；O 代表对输出到磁盘上的监视。这三个内容可以进行以下方式的组合，它们分别有七种不同的意义：

(1)	MON C	只对磁盘操作命令监视
(2)	MON I	只对由磁盘输入情形监视
(3)	MON O	只对输出到磁盘的情形监视；
(4)	MON C, I	监视磁盘命令和由磁盘的输入；
(5)	MON C, O	监视磁盘命令和向磁盘输出；
(6)	MON I, O	监视由磁盘的输入和向磁盘输出；
(7)	MON C, I, O	监视全部情形。

退出监视，可使用命令：

NOMON

类似于 MON 命令，NOMON 命令也有上面的七种用法。

监视命令可作为键盘直接输入命令立即执行；也可以用到程序的语句中去，MON 和 NOMON 命令作为 DOS 命令，它前面必须用控制字符 CTRL-D，如：

```
10  PRINT  CHR$(4); "MON  C, I, O"
```

135. 什么叫 EXEC 文件、它有哪些作用

顺序文件中的数据如果是一些特殊的命令资料，如 BASIC 命令和 DOS 命令，这种具有特殊数据的顺序文件称为 EXEC 文件。使用 EXEC 文件，要用 EXEC 命令，它的一般格式为：

EXEC EXEC 文件名

EXEC 文件有以下三种作用：

1.自动执行操作命令

在操作计算机时，经常要用到某些特定顺序的命令，例如，我们要进行以下操作:

HOME

RUN　PIC

LIST

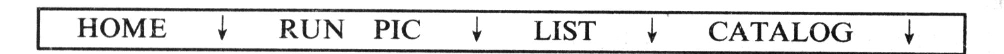
CATALOG

这些命令要反复使用，可以从键盘上顺序键入执行，也可以利用 EXEC 文件的形式来完成，方法如下:

先建立 EXEC 文件:

```
5   REM PROGRAM 135.1
10  D$ =CHR$(4)
20  PRINT D$;"OPEN FILE"
30  PRINT D$;"WRITE FILE"
40  PRINT "HOME"
50  PRINT "RUN PIC"
60  PRINT "LIST"
70  PRINT "CATALOG"
80  PRINT D$;"CLOSE FILE"
90  END
```

运行该程序后，在磁盘上建立了一个 EXEC 文件，名为 FILE，它的内容如下:



这时用 EXEC 命令运行 FILE 文件后，效果同在键盘上逐一键入是一样的。

这样，通过使用 EXEC 文件可使中华学习机自动地执行一批命令，简化了操作过程，提高了使用效率。

2. 程序的合并

EXEC 文件还有一种用法：就是建立一个 EXEC 文件，用 EXEC 命令把若干个程序文件合并。

例如，PRO1 是一个 BASIC 程序，其行号为 10～100，PRO2 是另一个 BASIC 程序，其行号为 200～500，现要将 PRO1 和 PRO2 合并成一个程序 PRO，其步骤如下:

(1)　键入 LOAD　PRO1

(2)　键入下面的程序

```
0  REM PROGRAM 135.2
1  D$ =CHR$(4)
2  INPUT "EXEC FILE NAME: ";F$
3  PRINT D$;"OPEN ";F$
4  PRINT D$;"WRITE ";F$
5  POKE 33,30
6  LIST 10,
```

```
7  PRINT D$;"CLOSE";F$
8  TEXT
9  END
```

这个小程序的作用是把 PRO1 变成程序文件，我们给它起名为 TPRO1（F$）。运行键入:

```
RUN
EXEC   FILE   NAME:   TPRO1
```

(3) 键入 LOAD PRO2

(4) 键入 EXEC TPRO1

(5) 键入 SAVE PRO

这时，磁盘中的 PRO 程序就是 PRO1 和 PRO2 的和并程序。

3. APPLESOFT 程序与整数 BASIC 程序之间的相互转换。

APPLESOF 程序转化为整数 BASIC 程序的操作方法如下:

(1) 装入该 APPLESOFT 程序，用上面介绍的方法，把它建立成程序文件，该程序文件起名为 TA。

(2) 在整数 BASIC 状态下（即">"），键入:

```
EXEC  TA
```

(3) 把程序中的 APPLESOFT 专有命令，改造成整数 BASIC 命令，存盘。

同样，对整数 BASIC 程序变成 APPLESOFT 程序的方法，依照上述步骤做相应的处理。

136. DOS 的引导步骤是什么

引导 DOS 就是把磁盘上的 DOS 装入主机内存，在主机内引导过程是分以下几个部分来完成的:

步骤 1: 由用户键入引导命令 PR#6，驱动器卡上的驱动子程序开始运行，这时将磁盘上的 0 道 0 扇区的引导程序 1 装入主机内存的 $800～$8FF 中。

步骤 2: 引导程序 1 开始运行，把磁盘上 0 道的 1～9 扇区中引导程序 2 和引导程序 1 一起装入主机内存 $3600～$4000 中。

步骤 3: 引导程序 2 开始运行，把磁盘上的 0 道 A～B 扇区的再分配程序与其余的 DOS 扇区装入主机内存引导程序的后面。

步骤 4: 运行再分配程序，把位于主机内 $1D00～$4000 存的 DOS 重新安置在 $9D00～$C000 中。

步骤 5: 运行 DOS 中的初始化程序，分配缓冲区，设定 HIMEM，设定 DOS 向量的指针，运行磁盘上的问候程序。

上述步骤如图 136-1 所示:

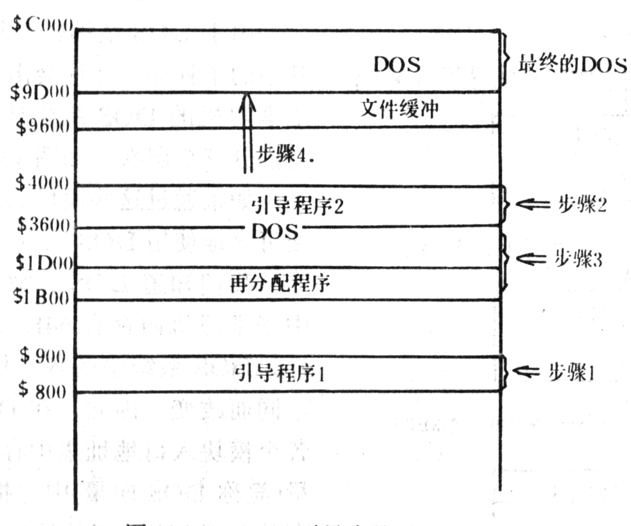


图 136-1　DOS 引导步骤图

137.　DOS 由哪几部分组成

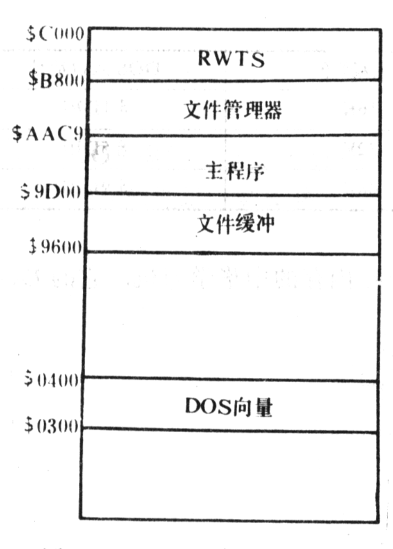


图 137-1　DOS 各部分位置图

引导 DOS 后，主机内的 DOS 有以下几部分组成：

(1) DOS 主程序：它的主要作用是实现 DOS 的初始化，与 BASIC 的接口，命令的解释，以及文件的管理等。

(2) 文件管理器：它主要是施行各个文件管理子程序，如 OPEN、WRITE、READ 等功能。

(3) DOS 向量：它包含对 DOS 主要子程序的调用命令、地址等。

(4) RWTS：是以扇区为单位进行读／写的子程序。

(5) 文件缓冲区：这是 DOS 读／写数据时使用的工作区域。

DOS 的各部分在内存中的位置如图 137-1 所示：

138. 什么是 DOS 向量、它有何作用

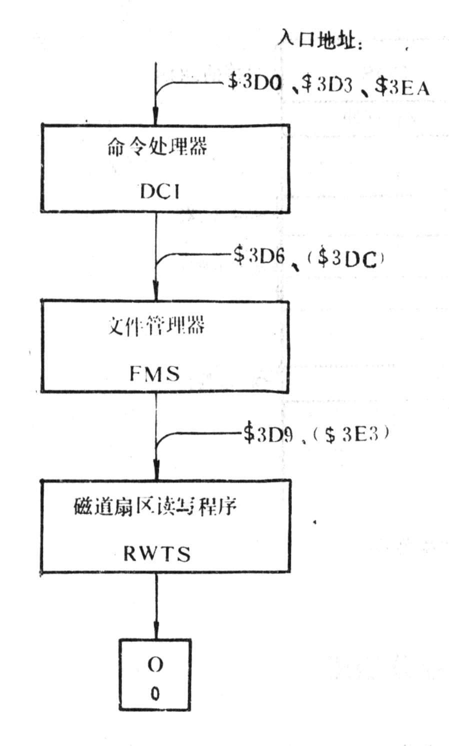


图 138-1 DOS 结构层次图

由于 DOS 结构较为复杂，直接调用或修改其中的子程序，往往会出现意想不到的错误。中华学习机的 DOS 系统是由模块化结构组成的，它们分三个层次，如图 138-1 所示:

如果通过这些模块的入口点来调用，则能方便可靠地使用 DOS 系统。

我们知道 DOS 存放在内存的最高地址区，中华学习机内存有 64K 等不同容量的内存。这样，DOS 系统中的各模块的入口地址会因内存不同而改变。因此，在 DOS 系统设计时，就把各个模块入口地址集中存放于第 3 页的 DOS 向量(简称 DOS 向量)中。把各个模块的入口地址存放于第 3 页的好处是，当使用不同内存的主机时，虽然 DOS 中各模块的入口地址改变了，但是 DOS 向量的位置却不改变，它的对应关系如下表所示:

表 138-1

向量位置	向量值	内存容量	DOS 入口地址
3D0G	$ 1DBF	16K	$ 1DBF
3D0G	$ 5DBF	32K	$ 5DBF
3D0G	$ 9DBF	48K	$ 9DBF

目前，大多数的用户均使用中华学习机。对于 48K 内存的中华学习机，它的 DOS 向量结构及作用如图 138-2 所示:

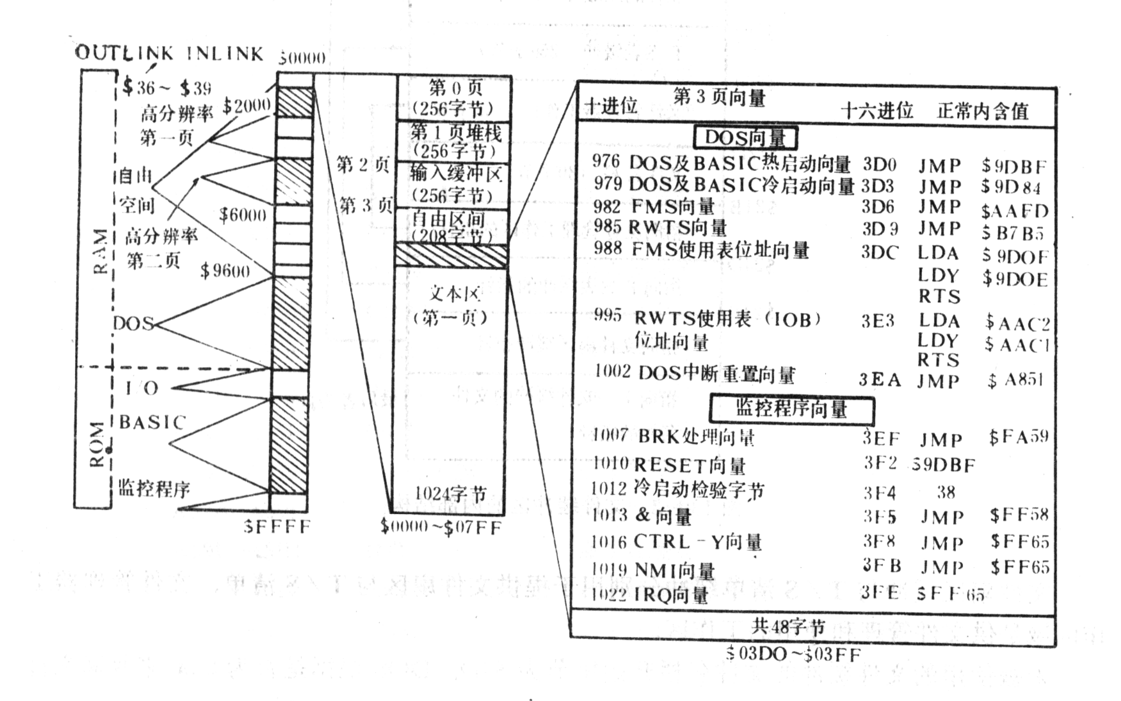


图 138-2　第 3 页向量的内存结构图

139. 什么是文件缓冲区

通常 DOS 提供了三个文件缓冲区，每个文件缓冲区占有 595 字节。文件缓冲区从 DOS 系统的开始地址（48K 内存，地址 $ 9D00），依次用指针链结。如图 139-1 所示：

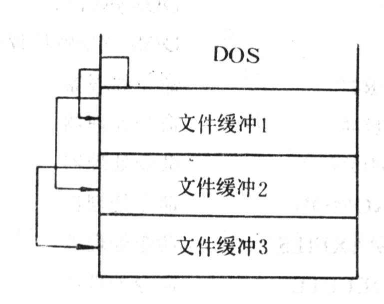


图 139-1　文件缓冲区

各文件缓冲区的内部结构如图 139–2 所示：

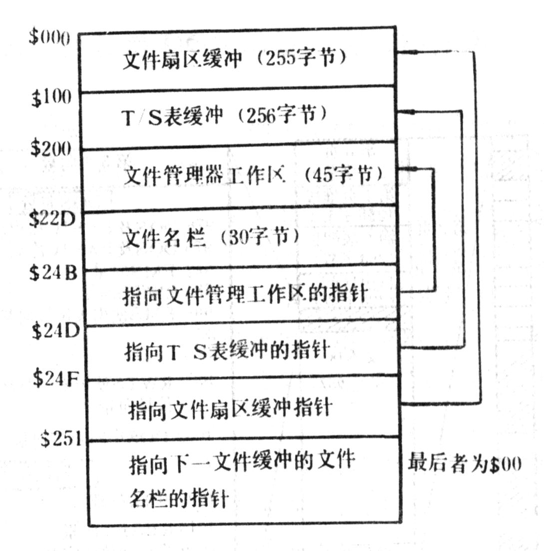


图 139–2　文件缓冲区的内部结构

文件扇区缓冲与 T／S 清单缓冲分别用于提供文件扇区与 T／S 清单，文件管理器工作区域是供文件管理和使用的工作区。

未被使用的文件缓冲的文件名栏开始字节为 $ 00。DOS 根据是否为 $ 00 来判断文件缓冲区是否正在使用，如果直接使用文件缓冲区时，最后必须要回到 $ 00。

140.　DOS 主程序有哪些作用

DOS 主程序在内存 $ 9D1E– $ AAB0 中，它包括 DOS 全部命令的处理子程序。详细分布如下。

9D1E–9D55		DOS 命令处理器内容项表
9D84–9DBE		DOS 冷启动内容项
9DBF–9DE9		DOS 热启动
9FCD–A179		DOS 命令解释程序
A229–A22D	PR#	命令处理器
A22E–A232	IN#	命令处理器
A233–A23C	MON	命令处理器
A23D–A250	NOMON	命令处理器
A251–A262	MAXFILS	命令处理器
A263–A270	DELETE	命令处理器
A271–A274	LOCK	命令处理器
A275–A27C	UNLOCK	命令处理器
A27D–A280	VERIFY	命令处理器

A281–A297	RENAME	命令处理器
A298–A2A2	APPEND	命令处理器
A2A3–A2A7	OPEN	命令处理器
A2EA–A2FB	CLOSE	命令处理器
A331–A35C	BSAVE	命令处理器
A35D–A38D	BLOAD	命令处理器
A38E–A396	BRUN	命令处理器
A397–A3D4	SAVE	命令处理器
A413–A479	LOAD	命令处理器
A4D1–A4E4	RUN	命令处理器
A4F0–A4FB	CHAIN	命令处理器
A510–A51A	WRITE	命令处理器
A51B–A525	READ	命令处理器
A54F–A56D	INIT	命令处理器
A56E–A579	CATALOG	命令处理器
A57A–A59D	FP	命令处理器
A59E–A5B1	INT	命令处理器
A5C6–A5DC	EXEC	命令处理器
A5DD–A60D	POSITION	命令处理器
A6A8–A6C3		命令管理器调用程序
A884–A908	DOS	命令名表
A971–AA3E		出错信息表
AA57	MAXFILES	
AA6D–AA6E	LOAD／BLOAD	长度
AA66–AA67		卷号
AA68–AA69		驱动器号
AA6A–AA6B		槽号
AA6C–AA6D		长度
AA6E–AA6F		记录
AA70–AA71		字节　命令参数
AA72–AA73		地址
AA74–AA75		MON 值
AA76–AA92		文件名缓冲
AA93–AAB0		供 RENAME 用的新文件名缓冲

141. 什么是 RWTS 子程序、怎样调用

RWTS 子程序是 DOS 的一部分，它的作用是读写磁道和扇区(Read or Write a Track and Sector)，也称磁盘驱动程序。

调用 RWTS 子程序需要先建立两个表，即 I／O 控制和设备特征表:

表 141-1　I／O 控制表(IOB)

字节($)	内　　　　容
0	表的类型　$ 01
1	槽号　x16
2	驱动器号
3	卷号
4	磁道号
5	扇区号
6、7	设备特征表的地址
8、9	用户数据缓冲区入口地址
A、B	未用
C	命令码(0-查找，1-读扇区，2-写扇区，4-格式化)
D	出错码
E～10	上次卷号，槽号 X16，驱动器号

表 141-2　设备特征表

字节($)	内　　　　容
0	设备类型　$ 00
1	每次步进数　$ 01
2.3	马达运转时间计数的补码，以 100 微秒为间隔 $ EFD8

按要求在内存中建立好两个表后，还需要写一个控制子程序，其内容是把 I／O 表的首地址放入 Y 寄存器(低位地址)和 A 寄存器(高位地址)中，然后转向 RWTS 子程序入口 $ 03D9 最后是子程序返回。

下面是一个如何使用 RWTS 子程序的例子:

将 IOB 表、设备特征表和控制子程序的实例全都放入内存的 $ C00 地址之后。控制子程序将 IOB 表的起始地址装入 A 和 Y 寄存器，然后跳转到 RWTS 子程序:

```
$ C00- A9 0C        LDA   # $ 0C      $ 0C″A
$ C02- A0 0A        LDY   # $ 0A      $ 0A″Y
$ C04- 20 D9  03    JSR   $ 03D9      转子程序 RWTS
$ C07- 60           RTS
$ C08- 00           BRK
```

在下面的 IOB 表中使用插槽号 6，驱动器号 1，把内存地址 $ 2000 开始的 256 个字节写入磁盘的 18 磁道的第 6 扇区。IOB 表如下:

```
$ C0A   01     $ C0E   12     $ C12   00     $ C16   02
$ C0B   60     $ C0F   06     $ C13   20     $ C17   00
$ C0C   01     $ C10   20     $ C14   00     $ C19   60
$ C0D   00     $ C11   0C     $ C15   00     $ C1A   01
```

设备特征表如下:

$C20　00　$C21　01　$C22　EF　$C23　D8

当上述准备工作完成后，可键入:

C00G 或 CALL 3072

将使整个程序执行。

142. 什么是磁盘的磁道和扇区

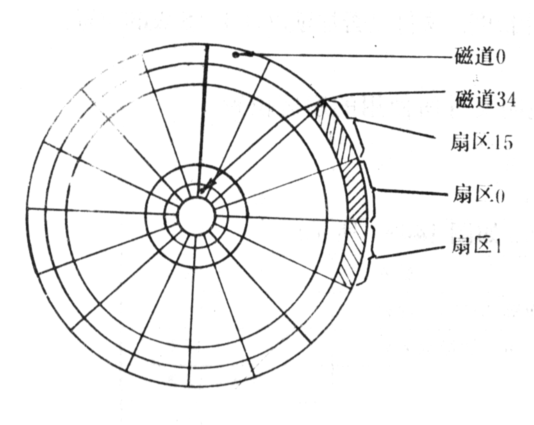


图 142-1　磁盘的磁道和扇区分布

每张磁盘被中华学习机的 DOS 分成 35 个同心圆，每个同心圆称为一条磁道，这些磁道编号为 0 到 34（$0 到 $22），其中最外层的磁道为 0 道，最内层的磁道为 34 道。

每条磁道又分成 16 段等弧，每段等弧称为一个扇区，扇区编号为 0 到 15（$0 到 $F），每个扇区能存储 256 个字节的资料，所以一张磁盘共有 560 个扇区共存储 143360 个（143K）字节的资料。磁盘的磁道和扇区分布如图 142-1 所示:

143. 磁盘初始化后的格式如何

磁盘初始化后，将磁盘划分成 16 个扇区，还将 DOS 写入磁盘的 0 道至 2 道，并把 17 道作为存储磁盘目录及其它管理信息的专用磁道，用户存放资料实际上只能使用 3 到 16 道和 18 到 34 道，共 31 条磁道 31 * 16 * 256 = 126976（字节）。初始化后的磁盘格式如下图所示:

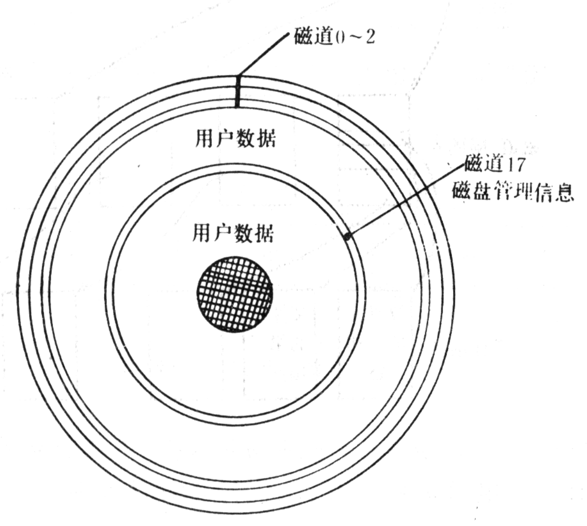


图 143-1　初始化后的磁盘格式

144. 什么是磁盘的管理信息

磁盘存储信息的基本单位是扇区，因此整张磁盘上所有扇区都是由以下四部分进行管理的:

(1) VTOC 表

VTOC (Volume Table Of Contents)表是用来提供整张磁盘的整体内部信息，如提供磁盘目录的位置、扇区的使用情况等。

(2) 磁盘目录扇区

用来记录磁盘中的文件名、文件类型、文件长度、文件是否加锁以及T／S表的位置。

(3) T／S清单表

T／S 清单表是磁道／扇区表，用来说明文件所使用的扇区位置。

(4) 文件扇区

记录程序内容及数据的扇区叫文件扇区。

以上四部分组成了磁盘管理信息的内容，如图 144–1 所示:

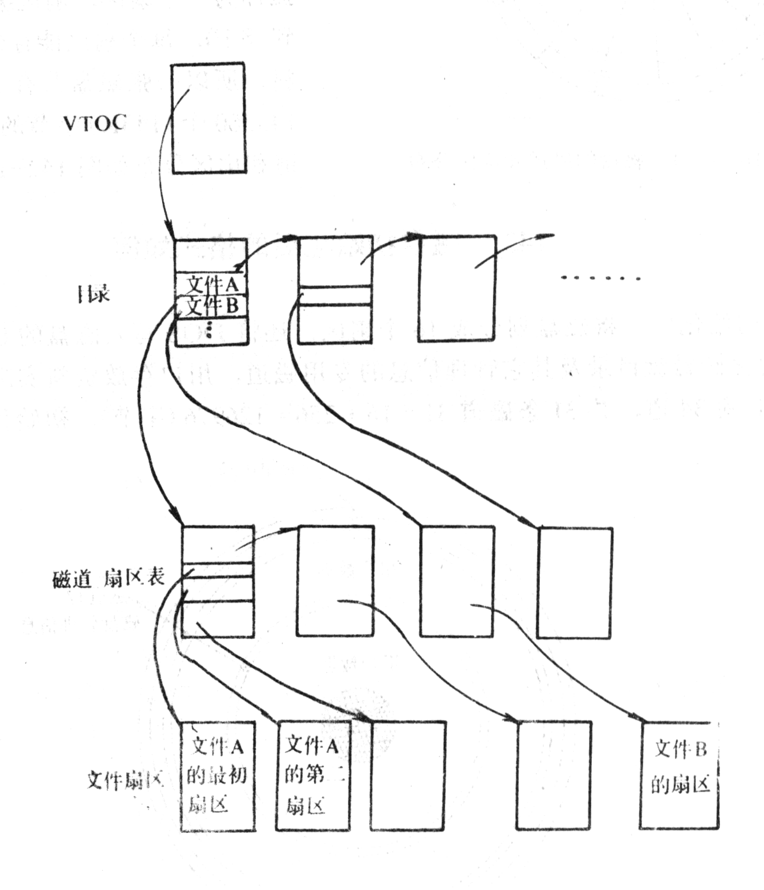


图 144–1 磁盘管理信息图

145. VTOC 表的结构是什么

VTOC 表占用一个扇区，它位于第 17 磁道的第 0 扇区。VTOC 表的内容如下表所示：

表 145-1

序　　号	用　途　及　内　容	
$ 00	未使用	$ 02
$ 01	第一个目录的磁道号	$ 11
$ 02	第一个目录的扇区号	$ 0F
$ 03	供 DOS 用	$ 03
$ 04～ $ 05	未使用	$ 00
$ 06	磁盘卷号，未指定时为 254	$ FE
$ 07～ $ 26	未使用	$ 00
$ 27	磁道扇区表的最大内容项数	$ 7A
$ 28～ $ 2F	未使用	$ 00
$ 30	最后一次写的磁道号码	$ FF
$ 31	改换磁道时，每次变换的磁道数目	$ FF
$ 32～ $ 33	未使用	$ 00
$ 34	磁盘中的磁道数	$ 23
$ 35	每磁道的扇区数	$ 0F
$ 36	每扇区的字节数(低位字节)	$ 00
$ 37	每扇区的字节数(高位字节)	$ 01
$ 38～ $ 3B	第 $ 00 道的磁道位图	
$ 3C～ $ 3F	第 $ 01 道的磁道位图	
$ 40～ $ BF	…	
$ C0～ $ C3	第 $ 22 道的磁道位图	
$ C4～ $ FF	未使用	$ 00

其中磁道位图是用来说明某条磁道上的扇区是否占用，下面举例说明磁道位图的内部结构，如图 145-1：

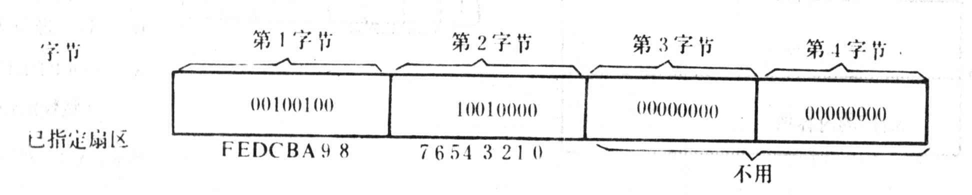


图 145-1　磁道位图的例子

在图 145-1 中，如果某位为 0，则表示相应扇区正在使用中；如果某位为 1，则表示相应扇区未被使用，图中的磁道上第 4，7，A，D 扇区未使用，其余的扇区均在使用中。

146. 目录扇区的结构是什么

目录扇区是用来管理存储在磁盘上的文件，目录扇区从磁盘的第 17 道的 15 扇区开始，根据磁盘上已存放的文件多少，可以一直链接到第 17 道的 1 扇区，如图 146-1 所示：

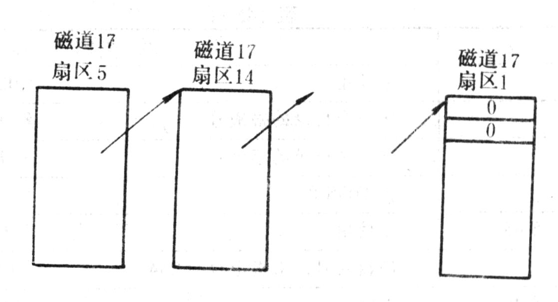


图 146-1　目录扇区的链接方式

每个目录扇区最多可记录 7 个文件，故整张磁盘最多可存放 105 个文件。每个目录扇区的结构如图所示：

目录	
$0	不用
$1	指向下一目录的指针（磁道号）
$2	指向下一目录的指针（扇区号）
$3 ~ $A	不用
$B ~ $2D	文件1的内容项
$2E ~ $50	文件2的内容项
$51 ~ $73	文件3的内容项
⋮	
$DD ~ $FF	文件7的内容项

图 146-2　目录扇区的结构　（一）

文件内容项相对字节	
$00	指向磁道 扇区清单指针（磁道号）
$01	指向磁道 扇区清单指针（扇区号）
$02	文件类型
$03 ~ $20	文件名
$21	文件的扇区数（低位）
$22	文件的扇区数（高位）

图 146-3　目录扇区结构　（二）

文件类型

L					B	A	I

L　1:LOCK
　　0:UNLOCK
B　1:二进制文件
A　1:APPLESOFT文件
I　1:整数BASIC文件
当B、A、I、都为0时是文本文件

图 146-4　目录扇区结构　（三）

147. T／S　表的结构是什么

磁道／扇区表即 T／S 表用于管理文件所使用的扇区位置。一张 T／S 表扇区最多可

以记录 122 个扇区位置。T／S 表所占用的扇区位置一般存放在磁盘的用户空间中。T／S 表的结构如图 147-1 所示:

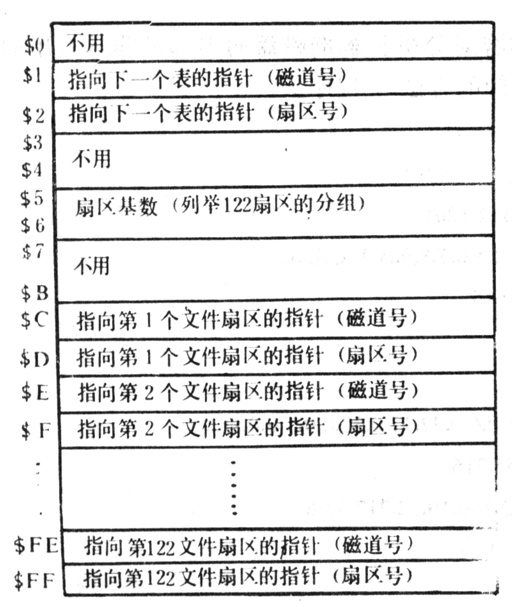


图 147-1 T／S 表的结构

148. 文件扇区的结构是什么

文件扇区是用来存放程序内容及数据的扇区，它与 T／S 表一样，都是存放在磁盘上的用户区域。DOS 系统检索磁盘上磁道的顺序是:

18 道， 19 道，…，34 道，17 道，16 道，…，3 道

而在各磁道上检索扇区的顺序为:

15， 14， 13，…，0

文件扇区的结构如图 148-1 所示:

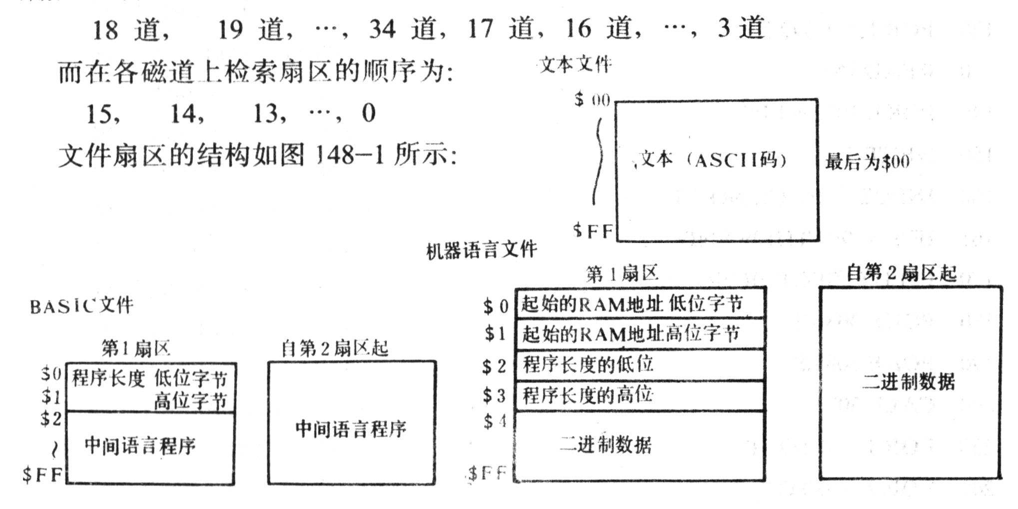


图 148-1 文件扇区的结构

注意：文件的类型不同，文件扇区的结构也就不同。

149. 如何显示磁盘扇区的内容

显示磁盘扇区的内容对分析和解剖磁盘的内部是非常有用的。显示的方法有很多种，这里我们介绍一种简易使的方法——程序法。

键入下面的程序:

```
0   REM PROGRAM 149.1
1   DIM A $ (20),H $ (20)
3   DATA 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
5   FOR I = 0 TO 15
7   READ A $ (I)
9   NEXT I
10  DATA 1,96,1,0,6,0,32,12,0,32,0,0,1,0,0,96,1
20  DATA 0,1,255,216
30  DATA 169,12,160,10,32,217,3,96
40  FOR I = 0 TO 16
50  READ IOB
60  POKE 3082 + I,IOB
70  NEXT I
80  FOR I = 0 TO 3
90  READ DC
100 POKE 3104 + I,DC
110 NEXT I
120 FOR I = 0 TO 7
130 READ INS
140 POKE 3072 + I,INS
150 NEXT I
160 INPUT "TRACE NO.";T
165 IF T = 999 THEN END
170 INPUT "SECTOR NO.";S
180 POKE 3086,T
190 POKE 3087,S
200 CALL 3072
230 FOR I = 0 TO 31
240 FOR J = 0 TO 7
250 N = PEEK (8192 + I * 8 + J)
260 GOSUB 500
270 H $ (J) = H $
275 IF N > 127 THEN N = N - 128
```

```
277  IF N < 32 THEN H $ (J + 9) = "."
280  IF N > 31 THEN H $ (J + 9) = CHR $ (N)
290  NEXT J
295  H $ (8) = ""
300  N = I * 8
310  GOSUB 500
320  PRINT H $ ;":";
330  FOR J = 0 TO 16
340  PRINT H $ (J);
350  NEXT J
360  PRINT
370  NEXT I
380  GOTO 160
500  REM DEC TO HEX
510  N2 = INT (N / 16)
520  N1 = N - N2 * 16
530  H $ = A $ (N2) + A $ (N1)
540  RETURN
```

运行程序后，输入要显示的磁道号和扇区号，程序就会自动显示该扇区的内容。其中屏幕左侧是扇区的十六进制内容，屏幕右侧是相对应的字符显示，当输入的磁道号为 999 时，程序运行结束。例如，要显示某张磁盘的第 17 道 15 扇区的内容，按照下面的提示进行：

```
TRACE NO.17
SECTOR NO.15
00  :00110E0000000000        ........
08  :000000130F82C8C5        ......HE
10  :CCCCCFA0A0A0A0A0        LLO
18  :A0A0A0A0A0A0A0A0
20  :A0A0A0A0A0A0A0A0
28  :A0A0A0A008001307        ....
30  :84C3C8C9CEC1A0A0        .CHINA
38  :A0A0A0A0A0A0A0A0
40  :A0A0A0A0A0A0A0A0
48  :A0A0A0A0A0A0A01A
50  :00150D82C2D2C9C4        ..BRID
58  :C7C5A0B1AEB0A0A0        GE 1.0
60  :A0A0A0A0A0A0A0A0
68  :A0A0A0A0A0A0A0A0
70  :A0A03000180D82C2        0....B
78  :D2C9C4C7C5A0B2AE        RIDGE 2.
```

```
80  :B0A0A0A0A0A0A0A0        0
88  :A0A0A0A0A0A0A0A0
90  :A0A0A0A0A032001B        2..
98  :0B02C2D2C9C4C7C5        ..BRIDGE
A0  :A0B2AEB3A0A0A0A0        2.3
A8  :A0A0A0A0A0A0A0A0
B0  :A0A0A0A0A0A0A0A0
B8  :32001E0980C2D2C9        2....BRI
C0  :C4C7C5A0D4C1C2CC        DGE TABL
C8  :C5A0DBB1DDA0A0A0        E [1]
D0  :A0A0A0A0A0A0A0A0
D8  :A0A0A009001E0080        .....
E0  :C2D2C9C4C7C5A0D4        BRIDGE T
E8  :C1C2CCC5A0DBB2DD        ABLE [2]
F0  :A0A0A0A0A0A0A0A0
F8  :A0A0A0A0A01100          ..
```

150. 如何更改磁盘扇区的内容

下面的程序有更改指定扇区的功能:

```
0  REM PROGRAM 150.1
1  DIM A $ (20),H $ (20)
3  DATA 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
5  FOR I = 0 TO 15
7  READ A $ (I)
9  NEXT I
10  DATA 1,96,1,0,0,0,232,12,0,32,0,0,1,0,0,96,1
20  DATA 0,1,255,216
30  DATA 169,12,160,210,32,217,3,96
40  FOR I = 0 TO 16
50  READ IOB
60  POKE 3282 + I,IOB
70  NEXT  I
80  FOR  I = 0 TO 3
90  READ DC
100  POKE 3304 + I,DC
110  NEXT  I
120  FOR I = 0 TO 7
```

```
130   READ INS
140   POKE 3272 +I,INS
150   NEXT  I
160   INPUT "TRACE NO.";T
170   INPUT "SECTOR NO.";S
180   POKE 3286,T
190   POKE 3287,S
200   CALL 3272
210   INPUT "BYTE POSITION?";B $
220   IF B $  = "" THEN 340
230   IF LEFT $ (B $ ,1) = " $ " THEN H $  = RIGHT $ (B $ ,2):GOSUB 600:B = N: GOTO 250
240   B = VAL (B $ )
250   IF B > 255 THEN 210
252   N = PEEK (8192 + B)
254   GOSUB 500
255   PRINT
256   PRINT "OLD DATA ";H $
260   INPUT "NEW DATA?";D $
270   IF D $  = "" THEN 210
275   IF N > 127 THEN N = N - 128
277   IF N < 32 THEN H $ (J + 9) = "."
280   IF LEFT $ (D $ ,1) = " $ " THEN H $  = RIGHT $ (D $ ,2):GOSUB 600:D = N:
 GOTO 300
290   D = VAL (D $ )
295   H $ (8) = " "
300   IF D > 255 THEN 260
310   POKE 8192 + B,D
320   B = B + 1
330   GOTO 252
340   POKE 3294,2
350   CALL 3272
360   END
370   NEXT  I
380   GOTO 160
500   REM DEC TO HEX
510   N2 = INT (N / 16)
520   N1 = N - N2 * 16
530   H $  = A $ (N2) + A $ (N1)
540   RETURN
```

```
600  REM HEX TO DEC
610  FOR I = 0 TO 15
620  IF A$(I) = LEFT$ (H$,1) THEN N1 = I
630  IF A$(I) = RIGHT$ (H$,1) THEN N2 = I
640  NEXT I
650  N = N1 * 16 + N2
660  RETURN
```

程序的用法如下

(1) 先输入要更改扇区的磁道号和扇区号(十进制数表示)。

(2) 接着输入扇区的相对字节位置(十进制数或十六制数表示)，若输入空格则程序结束。

(3) 显示由(2)所指定位置上的旧数据，输入新数据。这时便进到下一个字节上的数据位置，于是就可以继续输入新的数据了。若输入空格，程序就返回到(2)。

字节位置与新数据可以用十进制数或十六进制数形式输入，十六进制数必须加入"$"符号。

下面是一个操作实例:

```
    TRACK NO.5 ↓ →        输入磁道号与扇区号（十进
   SECTOR NO.6 ↓          制数）BYTE POSITION ? 5 ↓
→输入扇区内字节位置

                             OLD DATA 00 →
显示旧的数据                  NEW DATA ?65 ↓ →
输入新的数据（十进制数）

                             OLD DATA 00 →
显示下一个地址的数据          NEW  DATA  ?  $  23 ↓
→新的数据（十六进制数）

    OLD DATA 00 $         NEW DATA ? ↓
    BYTE POSITION ? $ 54 ↓

    OLD DATA 00
    NEW DATA ? $ 67 ↓

    OLD DATA 00
    NEW DATA ? $ 12 ↓

    OLD DATA 00
    NEW DATA ? ↓
    BYTE POSITION? ↓
```

151. 如何显示出正在使用的 T / S 表

下面的程序运行后可显示出文件使用的 T / S 表，以及文件扇区的磁道号与扇区号。

```
5   REM PROGRAM 151.1
10  DATA 1,96,1,0,17,15,176,13,0,32,0,0,1,0,0,96,1
20  DATA 0,1,255,216
30  DATA 169,13,160,154,32,217,3,96
40  INPUT "FILE NAME=";  F$
50  L = LEN (F$)
60  FOR I = 1 TO 30 - L:F$ = F$ +" ": NEXT I
70  FOR I = 0 TO 16
80  READ IOB
90  POKE 3482 + I,IOB
100 NEXT I
110 FB = 8192
120 FOR I = 0 TO 3
130 READ DC
140 POKE 3504 + I,DC
150 NEXT I
160 FOR I = 0 TO 7
170 READ INS
180 POKE 3472 + I,INS
190 NEXT I
200 CALL 3472
210 FOR I = 0 TO 6
220 J = 11 + I * 35 + 3 + FB
230 N$ = ""
240 FOR K = 0 TO 29
250 A = PEEK (J + K)
260 IF A > 128 THEN A = A - 128
270 N$ = N$ +CHR$ (A)
280 NEXT K
300 IF N$ = F$ THEN 380
310 NEXT I
320 T = PEEK (1 + FB)
330 S = PEEK (2 + FB)
340 IF T = 0 AND S = 0 THEN PRINT "FILE NOT FOUND":END
350 POKE 3486,T
360 POKE 3487,S
```

```
370  GOTO 200
380  T = PEEK (11+I * 35+FB)
390  S = PEEK (12+I * 35+FB)
400  POKE 3486,T
410  POKE 3487,S
420  PRINT "LS TRACK NO.=";T;" SECTOR NO.=";S
430  CALL 3472
440  FOR K = 0 TO 121
450  T = PEEK (FB+12+K * 2)
460  S = PEEK (FB+13+K * 2)
470  IF T = 0 AND S = 0 THEN END
480  PRINT "FS TRACK NO.=";T;" SECTOR NO.=";S
490  NEXT K
500  T = PEEK (FB+1)
510  S = PEEK (FB+2)
520  IF T = 0 AND S = 0 THEN END
530  GOTO 400
```

程序的用法如下:

(1) 输入文件名

(2) 显示文件的 T/S 表和文件扇区位置，其中 LS 表示 T/S 表扇区的位置，FS 表示文件扇区位置。

下面是一个实例，键入文件名 HELLO，显示问候程序的 T/S 表:

```
                            FILE NAME=HELLO
T/S 表扇区 <——              LS TRACK NO.=16 SECTOR NO.=15
文件扇区   <——              FS TRACK NO.=16 SECTOR NO.=14
                            FS TRACK NO.=16 SECTOR NO.=13
                            FS TRACK NO.=16 SECTOR NO.=12
                            FS TRACK NO.=16 SECTOR NO.=11
                            FS TRACK NO.=16 SECTOR NO.=10
```

本程序与 150 问介绍的程序合用，可以显示出文件在磁盘中的实际内容。

152. 磁盘目录磁道损坏后的补救方法

我们在使用磁盘的过程中，由于在目录磁道($11)上重复写入了其他数据，破坏了目录原有的信息，这种情况称为软损坏。有时，也会因为磁盘被弯折、擦伤、受潮等原因，可能使目录磁道所在部位受到损伤，这些物理损伤称为硬损伤。以上两种类型磁盘的损坏，使磁盘既不能装入文件到内存中，又不能把内存的程序文件存储到磁盘上，这样的磁盘变成了死盘或废盘。如果仅仅是目录磁道上的软损伤，可以启动 DOS3.3 盘，对该盘进

行初始化，使其仍旧具有装入和存储文件的功能。

下面我们着重介绍磁盘目录磁道物理损伤后的补救方法。

首先，对 DOS3.3 系统中的 INIT 和 CATALOG 等命令及相关程序的剖析，发现当引导 DOS 之后，下列有关地址中存放磁盘目录磁道的数据：

$AC01，$AE9F，$AEDC，$B293，$B3BC，$B4BC，$B5FA，$AEC5，$AEC9。

下列有关地址中存放磁盘格式化分成的磁道总数。

$AEB5，　$B3EF，　$BEFE

我们已经知道标准 DOS 磁盘的 $0～$2 道为 DOS 系统部分，$11 道为目录磁道，$12～$22 及 $3～$10 磁道为文件磁道。当磁盘目录磁道损伤后，我们重新设置一条磁道作为目录磁道。新设置的目录磁道最好取大于 $11 的附近数为好，磁盘格式化分成的磁道总数以 $1D 道左右为宜，因为这样具有可靠性。下面我们以 $14 为目录磁道，格式化分成磁盘为 $1C 道，这样使废盘避开了被损伤部分的 $11 道，使废盘在 $14 道的第 $0 扇区存入 VTOC 表，在 $1～$F 扇区存入文件名目录。具体方法如下：

1. 启动 DOS3.3 盘，NEW 清内存，CALL−151 进入监控。

2. 在下列地址中键入有关数据：

```
*AC01: 14    (目录磁道数)
*AE9F: 14
*AEDC: 14
*B293: 14
*B3BC: 14
*B4BC: 14
*B5FA: 14
*AEC5: 50    (5*目录磁道数)
*AEC9: 54    (4*目录磁道数+4)
*AEB5: 74    (磁道总数*4)
*B3EF: 1D    (磁道总数)
*BEFE: 1D
```

3. 用 CTRL−C 返回 APPLESOFT 状态后，换入废盘片，再键入一个 HELLO 程序。例如：

```
5  REM PROGRAM 152.1
10 VTAB 10: HTAB 16:PRINT "HELLO!"
20 VTAB 16: HTAB 20:PRINT "CEC−I DOS WORK DISK"
30 VTAB 20: HTAB 24:PRINT "1988.5.29"
```

如果光标再现，表示初始化成功。如果驱动器指示灯熄后，屏幕没有出现提示符及光标，这说明 $14 道附近也有损伤，格式化失败，可再另设其他磁道为目录磁道，将磁道总数削减得小一些，再进行试验。

153. 怎样用简易的方法测试磁盘的寿命

一张磁盘的使用寿命，主要由盘片的质量、驱动器的性能以及物理保护的方法来决定。但是，一张磁盘的使用寿命总是有限，如果等到盘片磨损到无法正确读写时再想到拷贝，已经为时过晚了。

为了物尽其用而又不过多过早地建立备份盘，有必要定期或时常测试一下磁盘的寿命。

下面我们介绍一种在中华学习机上以软件方式为主的简易测试方法。

先使中华学习机处于监控状态下，键入下列机器语言子程序：

```
0300- A9 20 8D C9 BD A9 52 8D
0308- CA BD A9 03 8D CB BD A9
0310- 7A 8D 6E BE A9 03 8D 6F
0318- BE A9 01 85 22 A2 00 BD
0320- 2B 03 9D 00 04 E8 E0 28
0328- D0 F5 60 C4 C1 D4 C1 A0
0330- D2 C5 D4 D2 C9 C5 D3 BD
0338- A0 A0 A0 A0 A0 A0 A0 D0
0340- D2 C5 D3 C5 CE D4 A0 D4
0348- D2 C1 C3 CB A0 A8 A0 A0
0350- A0 A0 A9 30 38 ED 78 05
0358- 48 4A 4A 4A 4A 09 B0 C9
0360- BA 90 02 69 06 8D 0D 04
0368- 68 29 0F 09 B0 C9 BA 90
0370- 02 69 06 8D 0E 04 CE 78
0378- 05 60 48 4A 48 4A 4A 4A
0380- 4A 09 B0 C9 BA 90 02 69
0388- 06 8D 24 04 68 29 0F 09
0390- B0 C9 BA 90 02 69 06 8D
0398- 25 04 68 4C 8E BE
```

键入命令把它存入磁盘：

BSAVE DISK TEST , A $ 300 , L $ 9E

通过使用 CALL 768 命令来调用它。程序运行后，屏幕顶端一行会显示。

DATA RETRIES =   PRESENT TRACK(   )

其中，等号后面的数目表示 DOS 在读取某一磁道上的资料时所作的尝试的次数（用十六进制数表示）。在正常情况下，尝试次数达到 15($ 0F)也是可能的，但是超过这一数目，就该考虑复制备份了。

另外，顶上一行括号中的十六进制数目，表示目前驱动器的读写头停留在磁盘的这一磁道上，当 CATALOG 时，这个数目是 $ 11(即十进制数的 17)，正是磁盘的目录磁道。

只要上述存储区 $ 300～ $ 390 的机器语言程序未改变，则测试功能始终有效。

154. 怎样快速地读取“T”型文件

在磁盘文件中，“T”型文件即文本文件，不能象“A”、“I”或“B”型文件那样用 LOAD 或 BLOAD 命令就能看到其中的内容，如果不知道文本文件的数据格式，就更难正确地读取。

为此，我们给你介绍一个万能的读取“T”型文件的程序，它能在屏幕上显示任何“T”型文件的内容，并做到正确无误。如果在下面程序加一个 PR#1 语句，就能在打印机上打印出来。

```
5 REM PROGRAM 154.1
```

```
10  REM * * * * * * * * * * *
15  REM *    RED TEXT     *
20  REM * * * * * * * * * * *
25  D $ = CHR $ (13) + CHR $ (4):P = 0: ONERR GOTO 85
30  HOME : VTAB 3: HTAB 12: PRINT "READ TEXT PROGRAM"
35  VTAB 5: NTAB 5: PRINT "DO YOU WANT CATALOG (Y / N)?";: GET YN $ :
 PRINT YN $
36  IF YN $ = "Y" OR YN $ = "N" THEN 38
37  GOTO 30
38  IF YN $ = "N" THEN 55
40  POKE 34,7: PRINT D $ ;"CATALOG": PRINT : PRINT "PRESS ANY KEY";:GET A $
55  VTAB 7: HTAB 5: INPUT "FILE NAME: ";C $ : IF ASC(C $ ) < 64 THEN 55
60  POKE 34,7: HOME
65  PRINT D $ ;"OPEN ";C $
70  PRINT D $ ;"READ ";C $
75  GET A $ :IF A $ = CHR $ (13)THEN PRINT CHR $ (0):B $ ;B $ = "":P = 1:GOTO75
80  B $ = B $ + A $ : GOTO 75
85  IF P = 1 THEN PRINT D $ ;"CLOSE ";C $ : END
90  IF PEEK (222) = 13 THEN PRINT "FILE TYPE MISMATCH": CHR $ (7):
 VTAB 22: HTAB 6: PRINT "PRESS ANY KEY TO INPUT AGAIN!";:
 GET A $ : POKE 34,0: HOME : RUN
95  IF PEEK (222) = 3 THEN PRINT "NO DISK IN DRIVE!"; CHR $ (7);:
 GET A $ : POKE 34,0: HOME : RUN
100  PRINT "FILE NOT FOUND!"; CHR $ (7): PRINT D $ ;"DELETE ";C $ :
 POKE 34,0: HOME : RUN
```

程序使用说明:

运行此程序后先问你是否要列磁盘目录，相应地按"Y"和"N"键。再键入想观看的"T"型文件的文件名，随后便在屏幕上显示出其内容。如果没有此文件或者文件类型不匹配或者驱动器中没有磁盘，就会显示出相应的错误信息，键入 CTRL−RESET 可使程序中断。

155. 如何将一些小程序藏在 DOS 之中

我们知道 DOS 系统占用了内存地址的 $ 9600 至 $ BFFF 区域的内容，在一般情况下，用户不得随便改动这部分区域的内容，否则，DOS 系统将被破坏。但是在这部分区域中，有些区域是空闲的，DOS 并未占用。如表 176−1 所示:

表 155-1

内 存 地 址	空 间 的 长 度
$ B6B3 − $ B6FC	73 个字节
$ BA69 − $ BA95	45 个字节
$ BC56 − $ BCFF	169 个字节

因此，用户可以将一些小程序藏在 DOS 中，用 JMP 指令连接起来。下面举一个应用实例：

在"FID"程序中，可用"＝"来代替文件名中的某些字符或文件名全体。例如：＝.OBJ 表示所有以.OBJ 结尾的文件名。

下面给出的机器语言子程序，可使 DOS 中也具有上述功能，只是用"−"来表示与之匹配的文件名。并且，我们将这段子程序藏在 DOS 系统中，即地址 $ BA69 至 $ BA81。然后，在 $ B201 中使用 JMP $ BA71 指令转向。

```
BA69.BA81
BA69- E8 B1 42 DD C6 B4 D0
BA70- 0A C8 C0 1E D0 F3 AE 9C
BA78- B3 18 60 C9 AD  F0 F7 4C
BA80- 0B B2
* B201.B203
B201- 4C 71 BA
```

156. 怎样把磁盘上的 B 型文件转储到磁带上

要将磁盘中的 B 型文件储存到磁带上，可以先用 BLOAD 命令把磁盘中的 B 型文件调入主机的内存单元中，然后再在监控状态下用 W 命令将内存单元中的机器语言或二进制数据存入磁带中。使用 W 监控命令的关键是知道 B 型文件存在内存单元中的起始地址和文件长度，而终止地址＝起始地址+文件长度。B 型文件存在内存单元的起始地址和文件长度可从相应的内存单元中查到，这些相应的内存单元地址根据系统容量大小而不同，请参看 161 问及表 161−1。

下面以 48K 系统为例说明将磁盘中的 B 型文件转储存到磁带上的两种方法。第一种方法：

(1) 将磁盘上的 B 型文件调入主机。键入：BLOAD 文件名

(2) 用 161 问的方法求出 B 型文件的起始地址和文件长度。

(3) 求 B 型文件在内存单元中的终止地址，终止地址＝起始地址+文件长度

(4) 将内存单元中的 B 型文件转存到磁带上。键下面的命令：

* 起始地址. 终止地址 W

这时先要按下录音机的录音键，再按下这个命令最后的回车键。完成上述四个部分就

将磁盘上 B 型文件存储到了磁带上了。

将磁带上的 B 型文件装入主机，要先键入下面的命令：

* 起始地址.终止地址 R

这时先按下录音机的放音键，再按下命令后的回车键，即可从磁带上调取 B 型文件。为了使从磁带中调取 B 型文件时，省去起始地址和终止地址，也可以按下面的第二种方法将主机内存中的 B 型文件转存到磁带上：

(1) 键入命令

* 200.2FFW 起始地址.终止地址 R 起始地址 G

(2) 先按下录音机的录音键，再按下回车键。

(3) 听到主机发出“嘟”的一声后，按下录音机的停止键(STOP)，使录音机停止工作。

(4) 键入命令

起始地址.终止地址 W

(5) 先按下录音机的录音键，再按下回车键。

如果要调用磁带中某个 B 型文件，用上面的方法存入后，可不必记住起始地址和终止地址，只要键入下面的命令即可：

* 200.2FFR

这时当听到两下“嘟嘟”声后，调取 B 型文件的工作结束。这种方法比前一种要简便得多。

157. 怎样建立一个“万能”的 RUN 命令

中华学习机 DOS 系统中共有三个执行文件的命令，它们是 RUN、BRUN 和 EXEC。对不同类型的文件要用不同的命令，使用起来比较麻烦。

我们给出下面的机器语言程序，把这三个命令合为一体。执行这个程序后，无论什么文件，执行时只需键入：

RUN　文件名

就能运行各种类型的文件了。

这段机器语言程序的设计思想是：先调用两个 DOS 子程序，一个是 $ A2A8(打开文件)，另一个是 $ A316 (关闭文件)，然后判断文件类型，分别执行相应程序。

另外，还可以把这段机器语言程序保存在磁盘上，键入：

BSAVE R，A $ 300，L $ 32

在磁盘的 HELLO 程序中加入下列语句：

```
999   PRINT CHR $ (4);"BLOAD R,A $ 300"
1000  CALL 768
```

这样，每次引导磁盘时，就具有上述 RUN 命令的功能。

机器语言程序如下：

```
0300- A9 DE 8D 24 9D A9 BC 8D          0310- 06 99 DF BC C8 D0 F5 60
0308- 25 9D A0 00 B9 18 03 F0          0318- 20 A8 A2 20 16 A3 A9 7F
```

```
0320- 2D C2 B5 F0 0A C9 04 D0          0330- C6 A5
0328- 03 4C 8E A3 4C D1 A4 4C
```

158. 怎样改进 BSAVE 命令

中华学习机上使用的磁盘操作系统 DOS3.3 中的命令 BSAVE，是用来把内存中的机器语言存入磁盘，它的格式是:

BSAVE f，Aa，Lj [，Ss] [，Dd] [，Vv]

其中参数 A 是机器语言程序的首地址，参数 L 是程序的长度。机器语言的首地址和末地址是很容易得的，只要列出程序清单即可，但程序的长度要经过计算得到。因此为了更方便地使用 BASVE 命令，可以用下面的程序修改标准的 BSAVE 命令格式的内容，使得 BSAVE 命令中只用到程序的首地址和末地址，而不用再计算程序的长度，修改的步骤如下:

1. 一张磁盘格式化后键入 CALL-151 命令进入监控状态。
2. 键入下列程序:

```
BCDF-08
BCE0-38 ED 73 AA 8D 6D AA 98
BCE8--ED 72 AA A8 C8 D0 03 EE
BCF0-6D AA AD 6D AA 28 20 E0
BCF8-A3 4C 54 A3
 * A964: FF
 * A351: 4C DF BC
```

3. 按下 CTRL-C 键回车，返回 BASIC 状态，用 INIT 命令重新格式化磁盘。这样，今后使用这张磁盘存储机器语言程序时，就可以使用修改后的 BSAVE 命令了，它的格式如下:

BSAVE f，Aa，Lj [，Ss] [，Dd] [，Vv]

其中 A 是程序的首地址，L 是程序的末地址。

159. 删除随机文件记录的简便方法

要在随机文件中删除记录，一般采用复盖法，即把需要删除记录后面的所有记录一个个向前移，复盖掉被删除的记录。例如：要删 3 号记录，则 0 至 2 号记录不动，把第 4 号记录往前移，写入第 3 号记录的位置，第 5 记录写入第 4 号位置，如此按顺序把以下的记录向前移。显然，被删除的记录位置越在前面，所移动的次数也越多，当记录数越多时，这种方法速度就越慢。

下面介绍一种简便方法，在记录号与序号无关时，可采用此法。

这种方法也是采用复盖法，即将最后的一条记录写入被删除记录的位置，同时把现有记录个数减 1。例如：某随机文件中共有 500 个记录，要删除第 90 条记录，那将第 500 号记录写入 90 号记录的位置，同时将记录个数减 1。这样，原第 500 号记录变为第 90 号

记录，现只有499个记录。

这种方法的优点是删除一个记录只需进行一次移动，速度较快。请读者不妨试一试。

160. 怎样知道磁盘的剩余空间

对于一个使用磁盘的用户来说，常常希望知道磁盘上还有多少空间可供用户使用。因为，把信息存入磁盘的过程中如果发生"DISKFULL"的情况，确实是一件麻烦事。

下面介绍一个工具程序，可以使用户在中华学习机上执行CATALOG时，看到磁盘的剩余扇区数，其显示的位置，就在原来显示磁盘卷号的地方。

以下是存放在$BA69到$BA8F间的机器语言子程序：

```
*BA69.BA8F
BA69- A9 00 85 40 85 41 A0
BA70- 8C 18 B9 F2 B3 F0 0E 0A
BA78- 41 68 18 90 F0 88 D0 E9
BA80- 90 FB 48 E6 40 D0 02 E6
BA88- A6 40 A5 41 20 24 ED 60
*BA69LL
    BA69-   A9 00          LDA   #$00
    BA6B-   85 40          STA   $40
    BA6D-   85 41          STA   $41
    BA6F-   A0 8C          LDY   #$8C
    BA71-   18             CLC
    BA72-   B9 F2 B3       LDA   $B3F2,Y
    BA75-   F0 0E          BEQ   $BA85
    BA77-   0A             ASL
    BA78-   90 FB          BCC   $BA75
    BA7A-   48             PHA
    BA7B-   E6 40          INC   $40
    BA7D-   D0 02          BNE   $BA81
    BA7F-   E6 41          INC   $41
    BA81-   68             PLA
    BA82-   18             CLC
    BA83-   90 F0          BCC   $BA75
    BA85-   88             DEY
    BA86-   D0 E9          BNE   $BA71
    BA88-   A6 40          LDX   $40
    BA8A-   A5 41          LDA   $41
    BA8C-   20 24 ED       JSR   $ED24
    BA8F-   60             RTS
```

把它存入磁盘，即键入：

BSAVE FREE SPACE.OBJ，A $ BA69，L $ 27

然后修改磁盘中的 HELLO 程序，加入下列语句：

```
1  REM PROGRAM 160.1 $ 67
5  REM * * * * * * * * * * * * * * * *
6  REM *   HELLO!!!  FREE SPACE!!  *
7  REM * * * * * * * * * * * * * * * *
10  POKE 44480,32: POKE 44481,105: POKE 44482,186
20  PRINT  CHR $ (4)"BLOAD FREE SPACE.OBJ"
30  HD $ = "FREE SPACE:"
40  FOR I = 1 TO 11: POKE 46011 - I, ASC ( MID $  (HD $ ,I,1)) + 128: NEXT I
```

这样，当用户使用这张磁盘启动时，就有了这项功能，例如：

```
]CATALOG

FREE SPACE: 484
*A 008 HELLO
*A 002 FREE SPACE
*B 002 FREE SPACE.OBJ
```

161. 怎样知道 B 型文件的起始地址和长度

磁盘中的一个 B 型文件，当用 BLOAD 命令将它装入主机内存后，有时需要知道 B 型文件在内存中的起始地址和长度，DOS 系统提供了下列单元地址存放上述信息，如下表：

表 161-1

RAM	起始地址		长度	
	低位	高位	低位	高位
48K	$ AA72(43634)	$ AA73(43635)	$ AA60(43616)	$ AA61(43617)
32K	$ 6A72(27250)	$ 6A73(27251)	$ 6A60(27232)	$ 6A61(27233)
16K	$ 2A72(10866)	$ 2A73(10867)	$ 2A60(10848)	$ 2A61(10849)

例如在 48K 系统的中华学习机中，B 型文件的起始地址和长度可用下面的立即执行命令显示。

PRINT PEEK(43634)+PEEK(43635) * 256　显示起始地址

PRINT PEEK(43616)+PEEK(43617) * 256　显示程序长度

用同样的方法可以显示 32K 和 16K 的系统中 B 型文件的起始地址和长度。

162. 如何加快装入文件的速度

磁盘使用中经常需要装入文件的操作，有时用 LOAD 命令装入一个较长的程序时，需要等十几秒的时间，甚至要等二十多秒。如果利用“快速装入文件程序”可提高速度 5 至 10 倍。

“快速装入文件程序”可作为磁盘的问候程序使用，方法如下：

(1) 引导原盘，键入 NEW 命令

(2) 将“快速装入文件程序”键入内存

(3) 用 DELETE HELLO 命令删除原盘上的问候程序。

(4) 用 SAVE HELLO 将“快速装入文件程序”作为新的问候程序存入磁盘。

如果用户的磁盘中 HELLO 程序另有用途，也可采用下面的方法：将“快速装入文件程序”作为一个独立的程序存入磁盘，再在该盘 HELLO 程序中的适当位置加上一条语句调用它。

“快速装入文件程序”的清单如下：

```
5 REM PROGRAM 162.1
10 FOR I = 0 TO 236
20 READ N
30 POKE I + 48815,N
40 NEXT I
50 DATA 169,0,76,67,190,173,201,181,141,108,191,141,117,191,173,202,181,
141,109,191,141,118,191,173,203,181,141
60 DATA 248,190,173,204,181,141,249,190,238,189,181,56,173,195,181,237,
189,181,133,66,173,196,181,233,0,133,67
70 DATA 172,189,181,174,193,181,138,208,8,173,194,181,240,83,206,194,
181,202,185,0,150,145,66,200,208,236,140
80 DATA 235,183,142,193,181,169,14,141,198,181,230,67,174,194,181,240,
81,165,67,141,192,181
85 DATA 32,73,191,238,192,181,206,194,181
90 DATA 208,245,173,192,181,133,67,173,203,181,141,191,181,173,204,181,
141,192,181,174,193,181,240,10,32,73,191
100 DATA 174,193,181,160,0,240,165,76,234,162,172,198,181,208,13,173,
201,181,174,202,181,160,1,32,101,191,160
110 DATA 12,173,191,181,174,192,181,32,101,191,96,141,240,183,142,241,
183,185,0,151,240,24,141,236,183,200,185
120 DATA 0,151,141,237,183,200,140,198,181,169,183,160,232,32,181,183,
176,8,96,32,234,192,162,5,208,5,32,234,162,162,8,104,104,104,104,138,76
,210,166
130 POKE 42103,76
```

```
140  POKE 42104,180
150  POKE 42105,190
160  END
```

163. 怎样显示被 DELETE 的文件名

我们知道磁盘上的文件用 DELETE 命令就可把它们删除，再使用 CATALOG 命令查看磁盘目录就不显示被删除的文件。键入下列语句：

POKE 44505，234

POKE 44506，234

再使用 CATALOG 命令就能显示出被删除的文件名。实际上是将＄ADD9(44505)设为 NOP。因为＄ADD9(45505)这个地址在 FMS 中，原指令为 BMI，作用是检查文件是否被删除。改为 NOP 之后，它就失去了作用。另外，也可将 BMI 改为 BPL。这样，CATALOG 时就只列出被 DELETE 的文件名。即：

POKE 44505，16

恢复 BMI 的方法如下：

POKE 45505，48

POKE 45506，74

不过，若被 DELETE 的文件名已被新文件名所覆盖，那么就不能显示出来了。

164. 如何增加存储机器码的长度

机器码用 BSAVE 命令存盘时，需要设定机器码的长度，即参数 L。由于 DOS3.3 系统中 BSAVE 命令的长度参数 L 限制在 32K 字节以内(＄7FFF)，但是在有些情况下，需要保存的文件超过 32K，可用下面的语句解除 DOS3.3 系统上的限制，使 B 型文件的长度放宽到 64K 字节，语句是：

POKE 43364，255

或修改磁盘中第 1 磁道 8 扇区中的＄64 单元内容，将这个单元的内容设为＄FF，即可增加存储机器码的长度。

165. 如何改变磁盘中问候程序的类型

在初始化磁盘时，一般把问候程序都定义为 HELLO 文件，文件类型只能是 APPLESOFT(A 型)。在实际应用中如果要更改 HELLO 的文件类型，如改为 B 型或 T 型文件，都是可以的，方法如下：

修改磁盘的第 0 磁道 13 扇区的＄42 单元，将此单元分别设置下列数值，其含义各有不同：

＄06　A 型文件 DOS 系统隐含值

＄14　T 型文件可用于 EXEC 文件

$ 34　B 型文件用汇编文件引导程序

166.　如何恢复被损磁盘中的 DOS 系统

磁盘的使用过程中，由于意外的事故磁盘中的 DOS 部分被损坏了，这时无法再引导这张磁盘。我们用下面的方法可以进行补救:

准备一张 COPY 软件将系统盘上第 0 道至第 2 道的内容拷贝到被损坏的磁盘上。这样被损坏 DOS 的磁盘就重新建立了 DOS，并能正常启动了，具体内容请参照 187 问的方法。

167.　如何增加磁盘的存储空间

有些磁盘是用来专门存放数据的，这类磁盘称为数据盘。数据盘中通常并不需要建立 DOS 系统，因此我们可以把 DOS 部分占用的扇区空间也放数据，从而扩大了数据盘的存储空间。这种方法的基本原理是：将数据盘 VTOC 表中第 $ 1 道和第 $ 2 道内容改写为 $ FF(表示为空磁道)，这样就使容量增加了二个磁道，共 32 个扇区。具体操作方法如下:

将数据盘放入驱动器，键入下面的程序，并且运行即可。

```
10 REM PROGRAM 167.1
100 REM * * * * * * * * * * * * * * *
101 REM * *     DOS KILLER ..... * *
102 REM * * * * * * * * * * * * * * *
110 TEXT : HOME
120 PRINT "DOS-KILLER": PRINT "————————": PRINT: PRINT "FREES 32 SECTORS
FOR AN ADDITICNAL 6%": PRINT "OF STORANGE,MAKING THE DISK
UMBOOTABLE": PRINT "ONCE A PROGRAM HAS OVERWRITTEN DOS."
140 PRINT : PRINT "INSERT DISK AND PRESS RETURN:";GET A $ : IF A $ < > CHR $
(13) THEN 260
160 POKE 47084,17: POKE 47085,0: REM SELSCT TRACK 17,SECTOR 0
170 POKE 47083,0: POKE 47091,0
180 LOC = 10000: POKE 47088,LOC - INT (LOC / 256) * 256: POKE 47089, INT
(LOC / 256): REM STORE DATA AT LOCATION 10000
190 POKE 768,32: POKE 769,227: POKE 770,3: POKE 771,76: POKE 772,217: POKE
773,3: REM ROUTING FOR JUMP TO RWTS
210 POKE 47092,1: CALL 768: REM READ SECTOR
220 POKE LOC + 60,255: POKE LOC + 61,255: POKE LOC + 64,255: POKE LOC + 65,
255 REM WRITE 4 BYTES
230 POKE 47092,2: CALL 768: REM WRITE SECTOR
240 VTAB 3: PRINT : CALL - 958
```

```
250  PRINT:PRINT "DONE.":PRINT:PRINT "UPDATE ANOTHER DISK?(Y/N):";:
 GET A$:IF A$ = "Y" THEN RUN
260  VTAB 20:END
```

168. 怎样创造一个新的 DOS 命令—TYPE

DOS 系统中，要检索一个文本文件必须编写一个检索程序才行。如果使用 CP/M 操作系统，可以用 TYPE 命令显示文件的内容，操作非常简单方便。在 DOS 系统中可以创造一个新的 DOS 命令—TYPE，让它完成显示文件内容的功能。方法如下：

(1) 键入机器码如下：

```
BC56- 20 A8
BC58- A2 20 8E FD 20 8C A6 09
BC60- 80 20 ED FD AD 00 C0 10
BC68- 17 2C 10 C0 C9 83 F0 17
BC70- C9 93 D0 0C AD 00 C0 10
BC78- FB 2C 10 C0 C9 83 F0 07
BC80- AD C5 B5 C9 05 D0 D5 20
BC88- 8E FD 4C FC A2
```

(2) 键入：

BSAVE BTYPE，A$BC56，L55

(3) 键入下面的 BASIC 程序：

```
5  REM PROGRAM 168.1
10  D$ = CHR$(4)
20  PRINT D$;"BLOAD BTYPE,A$BC56"
30  POKE 40222,85: POKE 40223,188
40  POKE 43140,84: POKE 43141,89
50  POKE 43142,80: POKE 43143,197
60  END
```

程序运行后，用 TYPE 命令就可以显示 DOS 系统下文件的内容了。注意：用 TYPE 命令显示 T 型文件以外的文件时，显示屏幕可能会出现一些奇怪的符号。

169. 怎样恢复被删除的文件

有时不注意错误地用 DELETE 命令删除了某个文件，在 DOS 系统中没有提供恢复文件的命令。下面给出的程序可将删除了的文件恢复。

先键入下面的程序：

```
5  REM PROGRAM 169.1
10  DATA 1,96,1,0,17,15,176,13,0,32,0,0,1,0,0,96,1
20  DATA 0,1,255,216
30  DATA 169,13,160,154,32,217,3,96
40  INPUT "DELETE FILE NAME?";F$
50 L = LEN(F$)
60  FOR I = 1 TO 29-L:F$ = F$ +" ":NEXT I
```

```
70  FOR I = 0 TO 16
80  READ IOB
90  POKE 3482+I,IOB
100  NEXT I
110 FB = 8192
120  FOR I = 0 TO 3
130  READ DC
140  POKE 3504+I,DC
150  NEXT I
160  FOR I = 0 TO 7
170  READ INS
180  POKE 3472+I,INS
19  NEXT I
200  CALL 3472
210  FOR I = 0 TO 6
220 J = 11+I * 35+3+FB
230 N$ = ""
240  FOR K = 0 TO 28
250 A = PEEK (J+K)
260  IF A > 128 THEN A = A-128
270 N$ = N$ +CHR$ (A)
280  NEXT K
300  IF N$ = F$ THEN 380
310  NEXT I
320 T = PEEK (1+FB)
330 S = PEEK (2+FB)
340  IF T = 0 AND S = 0 THEN PRINT "FILE NOT FOUND":END
350  POKE 3486,T
360  POKE 3487,S
370  GOTO 200
380  IF PEEK (FB+11+I * 35) < > 255 THEN PRINT "NOT DELETE FILE":END
390  POKE FB+11+I * 35, PEEK (FB+11+I * 35+32)
395  POKE FB+11+I * 35+32,160
400  POKE 3494,2
410  CALL 3472
420  END
```

然后运行程序，输入要恢复的文件名。根据文件的类型不同分别进行如下操作:

BASIC 文件(A 型):

键入：LOAD 文件名

DELETE 文件名

SAVE 文件名

机器语言文件(B 型):

键入: BLOAD 文件名

DELETE 文件名

BSAVE 文件名

文本文件(T 型)用 FID 程序复制。

上述操作完成后，就会在你的磁盘中恢复了被删掉的文件。

170. 忘了引导 DOS 怎么办

有时，我们在上机操作中花费了不少时间在键盘上打入一个较长的 BASIC 程序，正准备存盘时，突然发现还没有引导 DOS，而这时既不能将程序保存到磁盘上，又不能使用 PR#6 来重新引导 DOS，因为这样会把内存中的 BASIC 程序清除，应该怎么办呢?

引导 DOS 之所以会清除 BASIC 程序，是由于 DOS 在启动过程中使用了内存地址 $ 800 以后的 BASIC 程序使用区，而且做了许多初值设定工作。要保护内存中的 BASIC 程序不被清除，只要将程序在 DOS 引导之前先移入一个“安全地带”—内存地址 $ 5000~ $ 9000 的地方，要求程序的长度不超过这个范围，即使超过了也可以保护很大部分程序。然后引导 DOS，把程序再从 $ 5000~ $ 9000 单元移回到原位，这时我们会发现程序仍不能用 SAVE 命令保存到磁盘上，而且程序变得无法执行了。这是因为 SAVE、RUN 这些命令都要以零页中的一些位置来做指示器，而在 DOS 引导后便将这些指示器置了初值，所以程序就不执行命令了。因此，为了保护程序除了移动程序本身外，零页内容也要保护才行。下面是具体的操作步骤:

(1) 键入: CALL−151

(2) 记下零页中 BASIC 程序结束地址指示器($ AF, $ B0)中的内容。键入:

AF.B0

得到程序的结束地址。例如 $ AF−4A， $ B0−1F，则程序结束地址为 $ 1F4A，然后计算出程序的长度:

$ 程序场长度＝ $ 程序结束地址− $ 800+ $ 1

(3) 移动 BASIC 程序。键入:

5000<800.程序结束地址 M

(4) 保护 BASIC 的零页指示器。键入:

9060<60.FFM (假设程序不超过地址 $ 9060)

(5) 启动 DOS。键入:

C600G 或 6 CTRL−P

(6) 取回零页指示器。键入

60<9060.90FFM

(7) 移回 BASIC 程序。键入

800<5000.5000+程序长度 M

(8) 回到 BASIC 状态。键入

CTRL–C

(9) 用 SAVE 命令将 BASIC 程序存盘。

171. 怎样检测磁盘上损坏的磁道和扇区

检查磁盘的磁道扇区有无损坏，可以用特殊的工具软件完成，也可以用下面的一组程序完成。使用磁盘中，屏幕上出现错误信息“I／O ERROR”时，表示输入输出错误，这时，磁盘并无全部损坏，只要检查出该磁盘的哪些磁道扇区损坏，盘中的内容还可以保存。下面的一段机器语言运行后，可逐个将坏磁区的磁道号和扇区号以十六进制数显示在屏幕上，该程序设定磁盘为 40 磁道。程序分为三段机器码的地址为 $ 5000–$ 5016、 $ 5028–$ 5029、 $ 512A–$ 5287:

```
5000-20 58 FC 4C F6 51 00 06
5008-0C 02 08 0E 04 0A 01 07
5010-0D 03 09 0F 05 0B 00

5028-FF FF
512A-32 51 69 51 70 51
5130-83 51 C9 CE D3 C5 D2 04
5138-A0 C4 C9 D3 CB BB A0 A0
5140-D4 CB C5 CE A0 C8 C9 D4
5148-A0 D2 C5 D4 D5 D2 CE A0
5150-D4 CF A0 C3 C8 C5 C3 CB
5158-AC A0 CF D4 C8 C5 D2 A0
5160-A0 D4 CF A0 C5 CE C4 AE
5168-DE C5 D2 D2 CF D2 A0 DE
5170-A0 C6 CF D5 CE C4 A0 C6
5178-CF D2 A0 D4 D2 C1 C3 CB
5180-A0 DE A0 AC A0 D3 C5 C3
5188-D4 CF D2 A0 DE A9 02 D0
5190-02 A9 01 8D F4 B7 AD 28
5198-50 8D F0 B7 AD 29 50 BD
51A0-F1 B7 AD EC B7 30 24 C9
51A8-28 10 20 AD ED B7 30 1B
51B0-C9 10 10 17 A9 00 8D EB
51B8-B7 20 E3 03 18 20 D9 03
51C0-A9 00 85 48 B0 01 60 AD

51C8-F5 B7 60 A9 00 38 60 48
51D0-20 8E FD 68 0A A8 B9 2A
51D8-51 85 06 C8 B9 2A 51 B5
51E0-07 A2 00 A1 06 C9 DE F0
51E8-0C 20 ED FD E6 06 D0 F3
51F0-E6 07 4C E3 51 60 08 48
51F8-98 4B A2 10 BD E8 B7 9D
5200-17 50 CA 10 F7 A9 00 8D
5208-EC B7 8D 16 50 A9 00 20
5210-CF 51 AD 10 C0 AD 00 C0
5218-10 FB C9 8D F0 1C A2 10
5220-BD 17 50 9D EB B7 CA 10
5228-F7 AD 10 C0 A2 00 A9 8D
5230-9D 00 02 68 A8 68 28 4C
5238-03 E0 20 8E FD AC 16 50
5240-B9 06 50 8D ED B7 20 91
5248-51 90 20 48 A9 01 20 CF
5250-51 68 20 DA FD A9 02 20
5258-D4 51 AD EC B7 20 DA FD
5260-A9 03 20 D4 51AD ED B7
```

5268–20 DA FD 1 EE 16 50 AD 16

5270–50 C9 10 D0 C8 A9 00 8D

5278–16 50 EE EC B7 AD EC B7

5280–C9 28 D0 B9 4C 05 52 00

将三段机器码依次输入，按下 CTRL–C 回到 BASIC 状态，用命令：

BSAVE CD，A＄5000　L＄287

将机器码存入磁盘。为了便于使用上面的程序，我们再写入个 BASIC 程序：

```
10  PRINT CHR＄(4); "BLOAD CD，A＄5000"
20  CALL 20480
```

将此程序存盘，每次只要运行此程序，即可达到检测磁盘的目的。

运行程序后，屏幕上显示：

INSERT DISK，THEN HIT RETURN TO CHECK，OTHER TO END

表示插入需检测的磁盘，按下 RETURN 键开始检测，按其他键退出程序。

当磁盘损坏时，屏幕上会显示出：

ERROR 40 FOUND FOR TRACK ###，SECTOR###

TRACR 后是损坏的磁道号，SECTOR 后是损坏的扇区号。检查完毕，自动回到初始状态。

172. 主机的四种状态如何转换

中华学习机主机有四种不同的状态供用户使用，它们是：BASIC 语言状态、监控状态、小汇编状态、LOGO 语言状态。四种状态之间相互转换方式如下：

1. BASIC 语言状态与监控状态

由 BASIC 语言状态向监控状态转换键入命令：CALL −151

由监控状态向 BASIC 语言状态转换，如果不清除原内存中 BASIC 程序，按下 CTRL–C 键或键入命令：E003G；如果清除原内存中的 BASIC 程序按下 CTRL–B 键或 CTRL–RESET 键，还可以键入命令：E000G。

2. BASIC 语言状态与小汇编状态

由 BASIC 语言状态向小汇编状态转换，键入命令：CALL −11440(或 CALL 54096)

由小汇编状态向 BASIC 语言状态转换，键入命令：＄E000G

3. BASIC 语言状态向 LOGO 语言状态

由 BASIC 语言状态向 LOGO 语言状态转换，键入命令：LG 或 CALL 53516

由 LOGO 语言状态向 BASIC 语言状态转换，键入：BPT 后按下 CTRL–B 键；或者按下 CTRL–RESET 后再键入命令：E000G；或者按下 CTRL–RESET 后再按下 CTRL–B 键。

4. 监控状态与小汇编状态

由监控状态向小汇编状态转换，键入命令：D350G。

由小汇编状态向监控状态转换，键入命令：D360G 或＄FF69G

5. 监控状态与 LOGO 语言状态

由监控状态向 LOGO 语言状态转换清除原内存中的内容键入命令：LG 或 4000G。如果由 LOGO 语言状态转入监控状态的，那么由监控状态转回 LOGO 语言状态，不清除原内存

中内容，按下 CTRL–Y 键，或者键入命令：3F8H 或 1BFCG。

由 LOGO 语言状态向监控状态转换，键入命令：BPT 或按下 CTRL–RESET 键。

6. 小汇编状态与 LOGO 语言状态

由小汇编状态向 LOGO 语言状态转换，键入命令：$4000G 或 LG 或 $3F8G

由 LOGO 语言状态向小汇编状态转换，需多次按下 CTRL–RESET 键，直到屏幕上出现监控状态的提示符“*”，再键入 D350G

173. 如何改进 CATALOG 命令中文件占用扇区数的显示格式

我们知道，使用 CATALOG 命令后，磁盘的目录就会显示出来，其中每个文件所占用的磁盘扇区数采用十进制数方式显示其低位值。而当该文件占用的扇区数大于 255 时，在目录的扇区数位置上就会出现类似于“000.003”这样的值，很容易使 DOS 用户产生误会。因此，我们对 DOS 在内存中 $ADFC 至 $AE12、 $B78D 和 $B790 的地址内容，进行小小的修改，使 DOS 执行 CATALOG 命令时,文件占用扇区数以 4 位十六进制数显示。

例如：某张磁盘的目录在修改前显示出：

```
]CATALOG


DISK VOLUME 254
*A 002 HELLO
*A 009 COPYA
*B 003 COPY.OBJ0
*B 002 R1/2
*B 140 SUPER DISK COPY 3.6
*B 098 CHINA
*B 150 BCOPY
```

键入下列机器语言

```
*ADFC.AE12
ADFC- 20 8D B7 A9                AE08- DA FD BD E7 B4 20 DA FD
AE00- A4 20 ED FD BD E8 B4 20    AE10- 20 8D B7
*B78D.B78F
B78D- A9 A0 20
*B790.B792
B790- ED FD 60
```

再使用 CATALOG 命令后，则显示出：

```
]CATALOG
DISK VOLUME 254
*A $0002 HELLO
*A $0009 COPYA
```

```
* B $0003 COPY.OBJ0
* B $0002 R1/2
* B $008C SUPER DISK COPY 3.6
* B $0062 CHINA
* B $0096 BCOPY
```

同时，也可用 INIT 命令将此项功能保存到磁盘上。

174. 怎样打印中华学习机的命令文本

中华学习机的操作命令很多，用户有时总忘记几条，查阅有关书籍固然可以，但最好还是向机器询问，因为机器保存有它自己的命令文本。通过下面一段小程序可列出所有的 APPLESOFT 命令文本，错误信息文本。

程序如下：

```
5   REM PROGRAM 174.1
10  HOME
20  PRINT : PRINT "APPLE SOFT WORDS:": PRINT
30 I1 = 53456:I2 = 53855: GOSUB 1000
40  PRINT : PRINT "APPLE SOFT ERR TABLE:": PRINT
50 I1 = 53856:I2 = 54116: GOSUB 1000
60  PRINT "DOS COMMAND:": PRINT
70 I1 = 43140:I2 = 43271: GOSUB 1000
80  PRINT "DOS ERR TABLE:": PRINT
90 I1 = 43380:I2 = 43583: GOSUB 1000
100  END
1000  FOR I = I1 TO I2
1010 C = PEEK (I)
1015  IF C = 7 THEN PRINT " <BELL> ";
1016  IF C = 13 THEN PRINT " <CR> ";
1020  PRINT CHR$ (C);
1030  IF C > 128 THEN PRINT "";
1040  NEXT I: PRINT : PRINT
1050  RETURN
```

对于 DOS 命令及错误信息文本的修改，可利用“DISK DOCTOR”软件修改以下磁盘扇区的内容：$1 磁道 $07 扇区、$1 磁道 $8 扇区、$1 磁道 $9 扇区。

175. 如何在 DOS 启动时加入"欢迎辞"

一张磁盘在引导 DOS 的同时，如果能在屏幕上显示一行"欢迎辞"，那么其效果就更佳了。下面介绍的程序，就可以满足上述要求。

键入本程序后，使其运行。在光标的位置处，输入所需的"欢迎辞"，"欢迎辞"最长不超过 40 个字符。按下回车键后，所输入的"欢迎辞"就自动地在一行中央显示。这时，插入磁盘，再次按下回车键即可。以后，每次启动该盘时，屏幕都会显示出所设计的"欢迎辞"。

```
10  REM PROGRAM 175.1
100  HOME : HTAB 10: PRINT "BOOT MESSAGE MAKER": PRINT: PRINT "INPUT YOUR
MESSAGE USING 40 CHARACTERS OR LESS.THE MESSAGE WILL AUTOMATICALLY BE
 CENTERED.": PRINT : INVERSE : PRINT SPC( 40): PRINT : PRINT SPC( 40):
NORMAL
110  VTAB 22: PRINT " * * APPLE USER MONTHLY VOL.5 NO.26 * * ": VTAB 8:
 HTAB 1: INPUT "";M $ :L = LEN (M $ ): IF L > 40 OR L < 1 THEN 100
120  IF L < 39 THEN FOR X = 1 TO INT ((40 - L) / 2):M $  = " " + M $ : NEXT
 X: VTAB 8: HTAB 1: PRINT M
130  VTAB 13: HTAB 1: INVERSE : PRINT " INSERT TARGET DISK AND PRESS
RETURN ";: NORMAL : GET A $ : IF ASC(A $ ) < > 13 THEN 130
140  PRINT : PRINT : GOSUB 200
150  POKE 4103,32: POKE 4104,179: POKE 4105,8: FOR X = 4275 TO 4294:
READ Y: POKE X,Y: NEXT X
160  FOR X = 1 TO 40: IF LEN (M $ ) = > X THEN POKE X + 4297, ASC ( MID $
(M $ ,X,1)) + 128: GOTO 180
170  POKE X + 4297,160
180  NEXT X
190  POKE 788,2: GOSUB 220: PRINT : PRINT "BOOT MESSAGE SAVED.": END
200  FOR X = 768 TO 796: READ Y: POKE X,Y: NEXT X
210  DATA 169,3,160,8,32,217,3,96,1,96,1,0,0,0,25,3,0,16,0,0,1,0,0,96,1,239,216
220  CALL 768: IF PEEK (789) = 16 THEN PRINT : PRINT "UNERITE PROTECT DISK
AND  PRESS A KEY";: GET A $ : PRINT : GOTO 220
230  IF PEEK (789) = 64 THEN PRINT : PRINT "FIX DISK DRIVE ERROR AND PRESS
A KEY";: GET A $ : PRINT : GOTO 220
240  RETURN
250  DATA 32,88,252,162,0,189,202,8,157,128,5,232,224,40,208,245,165,43,74,96
```

176. 如何用磁带储存整张磁盘资料

为了确保磁盘上的资料，不因使用中的偶然原因而损坏，一般都是在使用前，先拷贝备

份盘。实际上，作为保存后备资料，磁带比磁盘要便宜得多。对于一个拥有相当数量软件或资料的机房来说，如能改用磁带代替目前用的磁盘来保存后备资料，就可以节省出很多磁盘，投入经常使用。通常使用磁带存储信息的方法是：一个文件一个文件地转录，这样的方法不仅手续麻烦，而且存入磁带时，每处文件都要录一段前导信号和校验信号，浪费大量磁带和时间。下面介绍一种利用中华学习机 DOS 中 RWST 子程序编制的转录程序，转录一张 5.25 英寸、35 条磁道的磁盘需要 15 分钟，所以如果用单声道收录机，一盘 C–60 磁带正好可以保存 4 面磁盘资料，如改用双声道收录机一个一个声道单独使用，则一盘磁带可以保存 8 面磁盘资料。

程序运行后，屏幕显示出：

DISKETTE < — > CASSETTE COPY

<1>　DISKETTE TO DISKETTE

<2>　CASSETTE TO DISKETTE

　　WHICH DO YOU WANT ? 1 OR 2

问你选择哪一项

键入 1，表示从磁盘向磁带转录。屏幕上立即显示：

PLEASE PUSH CASSETTE RECORDER <RECORD>

　　HIT ANY KEY TO RUN

这时按下录音机的 <RECORD>键后再按一下键盘上的任何键，屏幕上立刻显示：

　　IT'S RUNNING

表示转录工作开始。等待 15 分钟，屏幕上会出现：

　　FINISHED!

PLEASE STOP THE CASSETTE RECORDER

DO YOU WANT TO DO ANOTHER COPY?

告诉你这次转录工作已经结束，并且问你是否还有另一次转录。键入 Y 表示重新开始；N 表示程序结束。

键入 2，表示从磁带向磁盘转录。

程序如下：

```
5   REM PROGRAM 176.1
10  FOR I = 768 TO 794
20  READ M: POKE I,M
30  NEXT I
40  DATA 169,3,160,10,32,217,3,96,0,0,1,96,1,0,0,0,251,183,0,12,0,0,0,0,0,96,1
50  CALL-936: VTAB 8: HTAB 8: PRINT ″DISKETTE<—>CASSETTE COPY″
60  PRINT : PRINT ″<1> DISKETTE TO CASSETTE″
70  PRINT ″<2> CASSETTE TO DISKETTE″
80  PRINT : PRINT TAB( 8)″WHICH DO YOU WANT? 1 OR 2″
90  GET C
100  CALL-936: VTAB 12: PRINT ″PLEASE PUSH CASSETTE RECORDER″;
110  IF C = 1 THEN PRINT ″<RECORD>″
```

```
120  IF C = 2 THEN PRINT "<PLAY>"
130  PRINT: PRINT: PRINT: PRINT TAB(10)"HIT ANY KEY TO RUN": GET K $
140  CALL -936: VTAB 12: HTAB 13: PRINT "IT'S RUNNING"
150  POKE 790,C:T = 0:S = 0
160  FOR I = 1 TO 5
170  IF C = 2 THEN  GOSUB 320: CALL -259
180 L = 12
190  FOR J = 1 TO 112
200  POKE 782,T: POKE 783,S: POKE 787,L
210  CALL 768
220 L = L+1:S = S+1: IF S < 16 THEN 240
230 T= T+1:S = 0
240  NEXT J
250  IF C = 1 THEN GOSUB 320: CALL - 307
260  NEXT I
270  CALL - 936: VTAB 12: PRINT TAB (16);"FINISHED!"
280  PRINT: PRINT "PLEASE STOP THE CASSETTE RECORDER"
290  PRINT: PRINT "DO YOU WANT TO DO ANOTHER COPY?": GET A $
300  IF A $ = "Y" THEN 50
310  END
320  POKE 60,0: POKE 61,12: POKE 62,0: POKE 63,124: RETURN
```

177. 怎样使用软件 DOS TOOL KIT 中的 APA 程序

在“DOS TOOL KIT”这张工具箱磁盘中，有一个非常有用的程序 APA，它是“APPLE SOFT PROGRAMMER'S ASSISTANT”的缩写，其含义是“程序设计辅助工具”，顾名思义，它是一个从事 BASIC 语言程序设计的好助手，在我们工作中确实起到了很大的作用，现对它做些介绍与说明。

启动 DOS TOOL KIT 盘后，键入：RUN LOAD APA。这时，屏幕显示如下：

```
APPLESOFT PROGRAMMER'S ASSISTANT
VERSION 1.0
(C) COPYRIGHT 1979, APPLE COMPUTER INC.
&RENUMBER <START>,<INC>,<FIRST>,<LAST>
&HOLD
&MERGE
&LENGTH
&COMPRESS
&SHOW
&NOSHOW
```

&AUTO <START>,<INC>

&MANUAL

&XREF

&KEYS

此时，用户可毫无顾忌地使用如下命令:LOAD、RUN、SAVE、LIST、CATALOG等，不会影响“APA”的存在，但必须注意的是：FP,HIMEM,MAXFILES等命令决不能使用，并且如用CTRL−RESET,则会将APA中的&AUTO，&SHOW，&XREF的功能失效。在使用APA中的命令时，除&MANUAL的命令可用&MA代替外，其余皆可用&和命令的第一个字母代替。

1. &AUTO <START>,<INC>

这条命令能自动为你编行号。其中START表示起始行号，INC表示行号增加数。

例如：要写入一个程序让它从100行开始，间隔为5，可键入：

&A 100，5

按下RETURN。现在按下空格键，将出现行号100语句，然后按下RETURN键并按下空格键，下一个行号105即出现。

如果你希望留下一行不用，按RETURN键然后按空格键，下一个行号将出现，但没有键入语句。

如果输入有误，想消失该行号，再重新输入，则按CTRL−X，再按空格键，该行号重新显示。

2. &MANUAL

此命令是关闭自动编行号命令.键入：

&MA 按下RETURN键即可。

3. &RENUMBER <START>,<INC>,<FIRST>,<LAST>

此命令的作用是重新编排程序行号。其中FIRST表示重排的开始行号，LAST表示重排的最后行号。

例如：有一程序清单如下：

```
3  PRINT "T"
7  PRINT "A"
8  PRINT "I"
13  PRINT "P"
17  PRINT "I"
18  PRINT "A"
20  PRINT "O"
```

用TAI作为文件名来存盘。键入：

&R

你的整个程序将重新编号，从100行开始，其增量为10。即：

```
100 PRINT "T"
110  PRINT "A"
120  PRITN "I"
```

```
130  PRINT ″P″
140  PRINT ″I″
150  PRINT ″A″
160  PRINT ″O″
```

如果想指定起始行号与行号增量，可键入：

```
&R 10,5
```

这是，该程序的起始行号是 10，增加量为 5。

如果你只希望对程序的某一部分重新编号，你可以指定要改变的开始<FIRST>和结束<LAST>行号，例如：首先装入 TAI 程序，然后按下：

```
&R 100,10,7,13
```

其中，起始行号为 100，增量为 10，要重新编的开始和结束行号分别是 7 和 13，你将得到：

```
3  PRINT ″T″
17  PRINT ″I″
18  PRINT ″A″
20  PRINT ″O″
100  PRINT ″A″
110  PRINT ″I″
120  PRINT ″P″
```

4. &HOLD 和&MERGE

&HOLD 命令是把内存中的程序掩藏在 HIMEM 之上。这样它就不会被清掉。

&MERGE 命令是把内存中掩藏的程序取出。

&H 和&M 命令合用，可以使两个程序进行合并。例如，TIAN 程序如下：

```
100  PRINT ″TIAN″
110  PRINT ″HUA″
```

现在，我们要把 TAI 和 TAIN 两程序合并，操作如下：

```
LOAD TAI
&H
LOAD  TIAN
&M
LIST
  3  PRINT ″T″
  7  PRINT ″A″
  8  PRINT ″I″
  13  PRINT ″P″
  17  PRINT ″I″
  18  PRINT ″A″
  20  PRINT ″O″
  100  PRINT ″TIAN″
```

110 PRINT "HUA"

这样两个程序就合并成一个程序了。

5. &COMPRESS 命令

这条命令将消去内存程序中的 REM 语句，&COMPRESS 命令将让你保持两个程序版本，一个原编的一个是经压缩的。键入

&C

即可。

6. &LENGTH 命令

这个命令给出在内存的程序长度。使用时：按下：

&L

你将得到一个用十进制和十六进制表示的字节数，它就是内存程序的长度。

7. &SHOW

这个命令显示你程序中的控制字符。按下：

&S

控制字符将用反转形式显示出来。

8. &NOSHOW

这命令使控制字符不出现。按下：

&N

控制字符即不显示出来了。

9. &KEYS 命令

这个命令可以让你在键盘上增加三个特殊的控制字符。即“_”、“\”、“[”。键入此命令后：

&K

按下：

CTRL－O 表示 _

CTRL－L 表示 \

CTRL－K 表示 [

如果想删除&KEY 功能，可输入&MA 命令。

10. &XREF

这个命令可用来打印出程序中使用了哪些变量以及这些变量在哪些行号(语句)中出现。

例如：

```
10  T=50
20  I=T+30
30  PRINT I
&X
T   10,20
I   20,30
```

178. 什么是磁盘的同步码、它有何作用

我们知道，磁盘通过马达带动旋转，在磁头下略过，对应的磁道中必定以串引的方式存储着二进制数信息。由于磁道是闭合的圆周，这样磁道中储存的信息就成为无头无尾的二进制数信号流。为了能正确地读取磁道中信息，计算机必须要有一个办法来识别磁道中信息的起始位置，然后以每八个一组的方式把信号截取，送入计算机中进行处理。为此，必须在信号的前端置上同步码。在中华学习机中，同步码由连续 8 个“1”接着 2 个“0”共十个二进制数组成，同步码的个数至少要多于 5 个。借助于硬件与软件结合的方法对同步码进行识别。硬件是一串入、并出的转换器，把由磁头检测出来的串行信号送入移位寄存器，再平行输出。软盘中信息的格式规定最高位为“1”，如果最高位为“0”即被丢弃，而等到第一个二进制数为“1”时才能被接收。同步过程见图 178–1 所示:

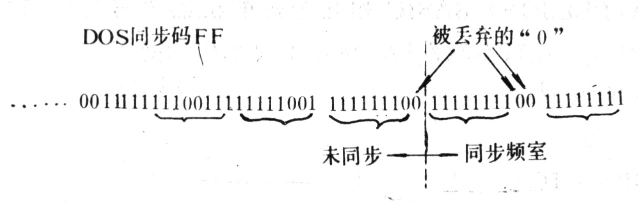


图 178–1　同步过程

计算机在调用磁盘中信息时，首先识别同步码，由于磁盘插入的位置是随机的，由图 178–1 知最多五次即可到同步的标识码“FF”，进入同步锁定。

利用改变同步码的方式可以实现软件的加密，如有的软件使用了“1110110000”的同步码，就能使某些拷贝程序无法复制。

179. 如何把 35 磁道的磁盘改为 40 磁道

标准的 DOS 3.3 系统格式化的磁盘为 35 磁道。但实际上，DOS 3.3 在 $ 11 磁道 $ 00 扇区的 VTOC 表上从 $ C4 至 $ FF 还保留了使用段落，供 35 以上的磁道使用，只是通常 INIT 以后把它们都做上已使用完的记号(填满 $ 00)，并在 VTOC 表的 $ 34 位置标上共有 35 个磁道。我们知道了这个原理后，就不难把它改成 40($ 27)磁道的磁道，从而增加 20K 的存储容量。

具体方法如下:首先把 DOS 调入内存。键入 POKE 48894,40($ BEFE:28)。在 1 号驱动器内将一张空磁盘格式化。然后运行下列程序，即可将这张磁盘改为 40 磁道。

程序如下:

```
1  REM PROGRAM 179.1
5  DATA 165,3,164,2,32,217,3,176,2,169,0,160,0,145,2,169,0,133,40,96,1,
96,1,254,17,0
10  FOR I = 768 TO 787: READ X: POKE I,X: NEXT I: POKE 2,232: POKE 3,183
20  FOR K = 47080 TO 47085: READ Y: POKE K,Y: NEXT K: POKE 47088,0: POKE
47089,96
```

```
30  POKE 47092,1: CALL 768: HOME : PRINT "PRESS ANY KEY TO CONTINUE":
 GET A $
40  FOR M = 24772 TO 24830": POKE M,255: POKE M + 1,255:M = M + 3: NEXT M
50  POKE 47092,2: CALL 768
```

180. DISK DOCTOR 软件的使用方法

DISK DOCTOR 软件又称磁盘医生。它可以将磁盘中的每个扇区的内容打印出来，并加以诊断，根据具体情况用不同的功能进行修改和检查。

本软件是 ZAP 实用程序的一种，它允许你正确显示修改磁盘的内容，且将资料写入指定的扇区中。这样你就可以利用该程序来修改磁盘目录，读出或写入不可显示的字符，使得程序得以加密或解密等等。

本软件由 APPLESOFT BASIC 和几个短的机器语言程序组成，机器语言主要是 BASIC 与 DOS 介面。运行本软件后，屏幕上显示出功能表：

```
* * * * DISK DOCTOR * * * *
        OPTIONS
    1)  DUMP SECTOR
    2)  ZAP SECTOR
    3)  RECOVER FILE
    4)  REMOVE DOS
    5)  SET DRIVE PARMS
    6)  END
ENTER SELECTION:
```

下面介绍各个功能的使用方法：

1. DUMP SECTOR

这项功能可以读出指定扇区的内容，并将其内容显示在屏幕上，或打印在纸上(若打印机型号不同，可修改 1120 语句，使得行宽不小于 80 个字符)。

2. ZAP SECTOR

此项功能可以修改磁盘中的任一字节。由于屏幕只能显示半个扇区内容，因此有转换显示功能。修改及控制键如下：

CTRL−M　移动模式(MOVE MODE)，按下此键后，可用 I，J，K，M 键将光标快速向右、左、上、下移动，以便将光标移到要修改的字节的地方。

CTRL−E　进入编辑模式(EDIT MODE)，允许进行修改，只要将光标移到要修改的地方，即可将十六进制数的资料写入目前资料项中，每次可修改半个字节，程序调用 ZAP 子程序将修改的内容显示出来。

CTRL−P 显示另一半扇区内容。

CTRL−W 将已修改的内容，写入磁盘。

CTRL−X 回到程序的主菜单上，不作任何修改。

以上五项功能可任意互换使用，来达到修改目的。

3. RECOVER FILE

此项功能可以使被误删除的文件复原。只要输入文件名即可，但要注意：修复之前，磁盘不能有文件的存入操作，而且在修复之后，须将文件拷贝到另一块盘上，而后再存入该盘，这是由于软件的工作过程决定的。在磁盘目录的 VTOC 表中，文件名并未删除，而仅仅被打上了删除标记，用此修改文件目录就可以对删除的文件进行 DOS 操作，但本软件并未在磁道位图上作标记，因此，被删除文件的扇区并未受到保护，随时有被清洗的危险，需要重新写入。

4. REMOVE DOS

此项功能是删除磁盘中的 DOS 系统，使磁盘具有更多的使用空间。

5. SET DRIVE PARMS

此项功能重新设置插槽号，驱动器号。

6. END

结束本程序。

181. 什么是 Pro DOS、 它与 DOS 3.3 有何异同

Pro DOS 是美国 APPLE 电脑公司 1983 年推出的 APPLE 机上最新的操作系统，要求有硬件设备：内存容量至少 64K 字节以上的主机，能够支持 128K 内存的结构，以及扩展 80 列文本卡。有了这些硬件上的支持，在你的命令下 Pro DOS 就能够进行很多操作了。中华学习机作为 APPLE Ⅱ微机的兼容机具备了要求的硬件设备，在中华学习机上同样可以开发 Pro DOS 操作系统。

Pro DOS 作为一个新型的操作系统，比 DOS 3.3 有更多的优异地方。它能支持在 APPLE Ⅱ或 APPLE Ⅱe 计算机及其兼容机上使用外加的 5 兆字节的硬盘；具有类似于 UNIX 操作系统的树状层次结构；文件种类可达 256 种；能够支持四种中断；支持时钟／日历卡。

Pro DOS 和 DOS 3.3 之间也有许多不同之处，DOS 3.3 的命令要指定驱动器槽号和磁盘号，而 Pro DOS 不用指定驱动器，而用卷号来指定磁盘，如果在主驱动器中找不到指定卷号的磁盘，它就会在系统所附着的所有的驱动器中去找，直到寻找到指定的卷号或一直寻找到最后一个驱动器为止。

从使用角度来看，Pro DOS 比 DOS 3.3 的读写文件快六倍左右。DOS 3.3 下写的程序能够在 Pro DOS 下运行，这是因为 Pro DOS 操作系统中，有 DOS3.3 与 Pro DOS 相互转换的管理功能程序，用户只要调用这一程序便可以实现数据或程序从 DOS 3.3 的盘上转换到 Pro DOS 盘上，反之亦然。

182. 在 ProDOS 下 APPLESOFT BASIC 有什么变化

从 DOS 到 ProDOS 有七个命令需要变化：

HIMEM 指针命令受到影响。因为 ProDOS 由 HIMEM 定出给新文件置的 I／O 缓冲区，ProDOS 以 256 字节一段来管理记忆体，因此 HIMEM 要经常下移 256，即 $

100。

INPUT 命令改变了。在 DOS 下 INPUT 会省去最后一个数据的“,”或“:”后的任何键入。如你键入字符串“ABCD,EFGH”去回答 INPUT A$ 你会发现 A$ 中仅有“ABCD”。ProDOS 下 INPUT 序列中最后一个串变量将会把所有字符进行保留，包括“,”和“:”。

TRACE 或 NOTRACE 是控制显示程序执行号的命令。在 DOS 下 TRACE 命令会影响 DOS 命令，在 ProDOS 下不会影响其它命令。

FRE 即属于 ProDOS 又属于 BASIC，但 ProDOS 的效率高。如跟有变量则是 BASIC 的，如在 PRINT 语句中且以 CTRL−D 开头或是在立即式中则是 ProDOS 的。ProDOS 的 FRE 与 BASIC 的 FRE(0)同样执行废料收集的功能。

APPLESOFT 在 ProDOS 下执行可能比 DOS 下稍慢一些，这是因为 ProDOS 监视着程序的每一行并去寻找 PRINT 语句第一个打印字符是否是 CTRL−D，ProDOS 以此来辨认命令。DOS3.3 也寻找 CTRL−D，但 DOS 不检查每一个执行的 BASIC 命令，DOS 仅寻找程序每个输出字符。由于操作系统在 ProDOS 下要比 DOS 快，程序的执行速度依然能在 ProDOS 下相对地说较快地执行。

183. 在中华学习机上怎样运行 ProDOS

ProDOS1.0.1 或 1.1.1 版的系统文件 PRODOS.SYS 里，有一段对机器商标进行检测的子程序，子程序经一系列的运算核对，若商标是“APPLE Ⅱ”就把 $0C 单元中的内容 $60 装入累加器，程序继续运行；若商标不是“APPLE Ⅱ”则向累加器装入 $00，终止程序运行。

要想在中华学习机上使用 ProDOS 操作系统，只要能在检测子程序运行完后，向累加器装入 $60，程序就可以运行下去了。因此我们可以有许多方法修改这段子程序。

(1) 从子程序开始 $2639 直接跳到 $265D，避开检测(键入 $2639:4C 5D 26↓)。

(2) 从子程序开始就把 $60 装入累加器，然后返回调主程序(键入: $2639:A9 60 60↓)。

(3) 把 $2661 单元的内容改成 $60，商标符不符送入累加器的都是 $60，程序也就自然会继续运行下去了。

下面以 PRODOS1.0.1 版为例说明修改的操作步骤:

(1) 启动 ProDOS 系统盘，按下 CTRL−RESET 键进入监控状态。

(2) 键入: *2639L↓列子程序的清单

(3) 修改键入: *2661:60↓

(4) 键入: *2000G↓当出现“STARTUP”的菜单，按 B 键退出。

(5) 键入: BLOAD PRODOS, A$2000, TSYS↓进入 PRODOS 的 BASIC 状态。

(6) 键入: CALL−151↓进入监控状态

(7) 键入: *2661:60 修改单元内容

(8) 按下 CTRL−C 键和回车键，退出监控状态。

(9) 键入: BSAVE PRODOS, A$2000, STYS

完成上述操作后，再次启动磁盘时，就会顺利地进入 ProDOS 操作系统。

对于 ProDOS1.1.1 版，只需把 $ 26A4 单元的内容改为 $ 60 即可，其它操作不变。

184. 如何制作 ProDOS 工作盘

在 DOS 3.3 磁盘操作系统中，要制作一张能开机引导的工作盘，可编写一个 BASIC 的引导程序，一般命名为 HELLO 程序，然后用 INIT HELLO 命令对磁盘作格式化处理，格式化后的磁盘上已存有 DOS 和 HELLO 程序，可作为工作盘来使用。但是 ProDOS 中没有 INIT 命令，也没有其他功能相同的命令可取代，磁盘格式化的工作必须利用系统盘中所提供的文件管理程序来完成，经 ProDOS 格式化后的磁盘并不能独立使用，因为这张磁盘上并没有存入操作系统和其它必要的文件，那么要制作一张可开机启动的 ProDOS 工作盘，必须具备 ProDOS 和 BASIC.SYSTEM 这两个系统程序，还应有 STARTUP 程序，下面所介绍的制作过程是在双驱动器的中华学习机上进行的。

1. 格式化工作盘，启动 ProDOS1.0.1 系统盘，在主功能表中，按 F 键进入文件管理程序，再按 V 进入 VOLUME COMMANDS，使用其中的格式化功能(FORMAT A VOLUME)给空盘作格式化处理，按下 F 键后，输入被格式化盘所在的槽号，驱动器号和工作盘号，也可以按三次回车键表示接受系统的约定值。

2. 用文件管理程序中的 COPY FILE 功能，将系统盘上的 PRODOS 和 BASIC.SYSTEM 拷贝到工作盘上，按两次 ESC，回到 FILER 功能表，按 F 键进入：FILE COMMAND，按 C 键选择 COPY FILE 功能，先拷贝 PRODOS，键入：PRODOS ↓，再键入：／工作盘卷号／PRODOS ↓，然后再按一次回车键，开始拷贝。完成后再同样的方法复制 BASIC.SYSTEM 程序。

3. 编写一个 BASIC 程序或从别的磁盘上拷贝一个程序，以 STARTUP 为文件名存入工作盘。STARTUP 程序中没有时钟卡，可由用户定时间。

4. 将其它程序存入工作盘，或将 DOS3.3 的程序转换过来。

这样就制成了一张 ProDOS 工作盘。

六、磁盘的加密、解密与复制

185. 如何对磁盘中文件名加密

编好一个程序文件保存在磁盘中，如果不希望别人看到这个程序，可以采用一种简单易行的文件加密方法，即在磁盘中的文件名中使用控制字符。

CTRL 键与字母键合用产生控制字符，它们在屏幕上不显示。文件名中使用了控制字符，用 CATALOG 命令是看不出来的，对于不知道的人来说，就不易从磁盘中把文件装入或运行，从而起了加密作用。

注意，文件名中的控制字符不能使用 CTRL−C、CTRL−M、CTRL−H 和 CTRL−U。用→键复制有控制字符的文件名时，控制字符不被复制。

186. 如何隐去磁盘中的文件名目录

如果能把磁盘中的文件名目录的一部分或全部隐藏起来，使 CATALOG 命令失效，这样也可以达到文件加密的作用。

一、隐藏全部文件目录的步骤:

1. 引导 DOS，插入要加密的磁盘。

2. 键入下列小程序。

```
5   REM PROGRAM 186.1
10  FOR I = 833 TO 872 : READ N : POKE I,N: NEXT I
20  CALL 833
30  POKE 24587,0
40  POKE 859,2
50  CALL 833
60  END
100 DATA 169,3,160,79,32,217,3,169,0,133,72,96,0,0,1,96,1,0,17
110 DATA 15
120 DATA 101,3,0,96,0,0,1,0,0,96,1,0,0,0,0,0,0,1,239,216
```

3. 运行 RUN

这时，用 CATALOG 命令不能显示目录，而只有“DISK VOLUME 254”的信息。如果要恢复磁盘目录的显示，可将上述程序的 30 语句改为:

```
30  POKE 24587,18
```

再运行本程序即可。

二、隐藏部分文件目录的步骤:

有时只须隐藏磁盘目录的一部分，例如只显示最前面的七个文件名，可在上述程序中加入:

```
25  POKE 24578,0
```

操作方法如下:

1. 磁盘初始化后，存入六个不需隐藏的文件，或存入六个毫无内容的“程序”。
2. 存入要隐藏的文件。
3. 键入上述程序，并加入 25 语句。
4. 运行 RUN

若要恢复原盘，则需要修改下述两语句:

```
25  POKE 24578,14
30  POKE 24587,18
```

有趣的是，此时如果将 25 语句修改为:

```
25  POKE 24578,15
```

则 CATALOG 后，前面七个文件名无限循环而不能显示其它文件名。

三、自动隐名与自动复原的步骤:

如果要计算机自动把隐目录盘复原，自动装入隐文件名投入运行，然后又自动隐去部分目录文件。可采用下述方法:

1. 要在目录中隐去文件名的程序(假设文件名为 PRO)的第一行加入下面的语句:

```
0  POKE 24578,0:POKE 24587,0:CALL 833
```

2. 按二.的 1、2 步骤操作。
3. 将隐藏部分文件目录程序的 60 语句改为:

```
60  PRINT CHR $ (4);"RUN PRO"
```

并把它作为问候程序(HELLO)。

4. SAVE HELLO
5. PR#6

等驱动器灯灭后取出磁盘，就行到了隐目录盘。以后每次启动该盘时，都能自动装入 PRO 程序并运行，同时将目录隐去一部分。

注意：该种加密磁盘不能写保护。如果对 BASIC 程序加密效果更佳。

187. 如何复制磁盘上任意磁道任意扇区的内容

在前面 171 问中曾讲到可以检测磁盘坏的磁道和扇区的位置，如果这张盘上其它部位的信息并未损坏，可以复制到另外一张磁盘上继续使用。下面介绍一种方法就可达到此目的，步骤如下:

1. 键入下面的 BASIC 程序，并运行。

```
5  REM PROGRAM 187.1
10  DATA 1,96,1,0,6,0,32,3,0,32,0,0,1,0,0,96,1,0,1,255,216,16,9,3,160,10,
32,217,3,96
14  FOR I = 0 TO 16 : READ IOB : POKE 778 + I , IOB : NEXT I
```

```
15  FOR I = 0 TO 3 : READ DC : POKE 800 + I , DC : NEXT I
16  FOR I = 0 TO 7 : READ INS : POKE 768 + I , INS : NEXT I
18  RWTS = 768 : TRK = 782 : SEC = 783 : SLOT = 779 : DISK = 780 :
WR = 790
20  X $ (1) = ″DISK DRIVE SLOT″ : X $ (2) = ″SOVRCE DRIVE″ :
X $ (3) = ″DESTINATION DRIVE″
22  X $ (4) = ″STARTING TRACK″ : X $ (5) = ″ENDING TRACK″ :
X $ (6) = ″STARTING SECTOR″ : X $ (7) = ″ENDING SECTOR″
25  Y $ (1) = ″06″ : Y $ (2) = ″01″ : Y $ (3) = ″02″ : Y $ (4) = ″00″ :
Y $ (5) = ″34″ : Y $ (6) = ″00″ : Y $ (7) = ″16″
27  W $ (1) = ″7″ : W $ (2) = ″2″ : W $ (3) = ″2″ :W $ (4) = ″40″ :
W $ (5) = ″40″ : W $ (6) = ″15″ : W $ (7) = ″15″
28  X $ (8) = ″SOURCE″ : X $ (9) = ″DESTINATION″
30  TEXT : HOME : FOR I = 1 TO 7 : IF I > 5 AND X(4) < > X(5) THEN 40
32  VTAB I : HTAB I : PRINT X $ (I) ; ″->″ ; : INVERSE : PRINT Y $ (I) ; :
INPUT ″″ ; Z $  : IF Z $  = ″″ THEN Z $  = Y $ (I)
34  IF Z $  < ″0″ OR Z $  > W $ (I) THEN NORMAL :GOTO 32
36  X(I) = VAL ( Z $ ) : VTAB I : HTAB 22 : PRINT X(I) ; ″″ : NORMAL :
IF I = 5 AND X(5) < X(4) THEN 32
38  IF I = 7 AND X(7) < X(6) THEN 32
40  NEXT I : POKE SLOT , X(1) + 16
45  VTAB 9 : IF X(2) < > X(3) THEN PRINT ″PRESS RETURN KEY CONTING″ ; :
GET A $
50  FOR I = X(4) TO X(5) : IF X(4) < X(5) THEN X(6) = 0 : X(7) = 15
52  FOR J = X(6) TO X(7) : FOR K = 1 TO 2 : IF X(2) = X(3) THEN VTAB 9 :
HTAB 1 : PRINT ″INSERT″ ; : INVERSE : PRINT X $ ( K + 7 ) ; : NORMAL :
PRINT ″DISK AND PRESS RETURN″ ; : GET A $
55  POKE TRK , I : POKE SEC , J : POKE DISK , X( K + 1 ): POKE WR , K :
CALL RWTS
60  NEXT K : NEXT J : NEXT I : GOTO 30
```

2. 程序运行后,屏幕显示需用户回答的参数:

DISK DRIVE SLOT : (驱动器槽口)
SOURCE DRIVE : (原盘所在驱动器号)
DESTINATION DRIVE : (复制盘驱动器号)
STARTING TRACK : (开始复制的磁道号)
ENDING TRACK : (终止复制的磁道号)
STARTING SECTOR : (开始复制的扇区号)
ENDING SECTOR : (终止复制的扇区号)

这里需给出的磁道号和扇区号均以十进制数表示。

3. 选择好参数后，按屏幕上的提示就可复制了。

这样，任意磁道任意扇区的内容就可以复制到另一张盘上了。

188. 怎样修改扇区号对磁盘加密

磁盘的加密方法很多，原理各不相同，下面介绍的方法是利用了DOS中逻辑扇区号与物理扇区号对应的关系：

物理扇区：0123456789ABCDEF

逻辑扇区：0DB97531ECA8642F

只要改变上述对应关系：就可得到一张加密的磁盘，操作步骤如下：

1. 将要加密的磁盘按通常方式用DOS格式化

2. 输入下面的程序：

```
5  REM PROGRAM 188.1
10  TEXT :HOME:DIM A(15): IV = 11 * 4996+ 15 * 256+11 * 16 +8
100  FOR I=0 TO 15 :A(I)=PEEK(IV+I):NEXT
110  C=INT (RND (1) * 15+1):B=1+INT (RND(1) * 15)
120  T=A(B):A(B)=A(C):A(C)=T
130  GET A $ :IF A $ ="Y" THEN 110
135  IF A(15)=15 THEN T=A(15):A(15)=A(8):A(8)=T
140  FOR I=0 TO 15 :POKE IV+I ,A(I):POKE 46669+I,A(I):NEXT
```

3. 运行上面的程序：当出现闪烁光标时，按下Y键可使程序再运行，按下其它键则程序停止运行。

4. 程序运行结束后，用NEW命令清内存。

至此，这张盘就是一张加密的盘了，只有用该密盘的DOS才能运行盘上的文件。但是要注意的是，当上面的程序运行的次数不同时，其DOS也是不同的，也就是不能通用。

通常可以用加密后磁盘的DOS格式化若干张盘，然后再将空盘的DOS部分去掉，这样存入文件后，就只能用原加密盘才能运行文件，达到了简单保密的目的。

189. 怎样修改DOS命令

所有的DOS命令都是在系统启动时由磁盘装入的，我们可以修改CATALOG、BLOAD、RUN、BRUN等命令，使非法用户看不到磁盘上的文件名，也不能将程序装入或运行。修改DOS命令可以起到磁盘加密的作用，下面介绍两种修改DOS命令的方法。

1. 直接修改法

DOS操作系统中，有一个DOS命令名称清单表。每一个DOS命令都是以ASCII码形式存放在内存中，地址在 $ A884～ $ A907(43140～43272)之间。这段地址中存放了28条DOS命令的ASCII码。其中，每一条命令除了最后一个ASCII码的最高位为1(即ASCII码大于 $ 80)，其它的最高位都为0(即ASCII码小于 $ 80)。例如，INIT为 $

49(I), $ 4E(N), $ 49(I), $ D4(T)。

修改 DOS 命令，只需改变相应命令的 ASCII 码。例如，让 TAI 来代替 CATALOG，操作如下：

```
CALL-151
*A8D2:54 41 C6
*A8D5<A8D9.A907M
*FP
INIT  HELLO
```

经过这样的处理，可得到一个加密的空白磁盘。同理，可举一反三，修改所有的 DOS 命令。

注意：修改后的 DOS 命令的字母个数，不易超过原来命令的字母个数。若个数小于原来的个数，应将其后命令依次靠拢。否则，会引起命令的混乱。

2. 程序修改法

根据上述的原理，利用 BASIC 程序来修改 DOS 命令。程序如下：

```
10  REM PROGRAM 189.1
110  DIM C$(28)
120  TEXT : HOME : VTAB 4: GOSUB 1000
130  VTAB 20: HTAB 5: PRINT "NO.<1...28>";: INPUT N
140  IF N < 1 OR N > 28 THEN 130
150  VTAB 21: HTAB 1: PRINT "NO.";N;":"; TAB ( 20);"-->";: INPUT C
160  IF LEN (C$) > LEN (C$(N)) THEN C$ = LEFT$ (C$, LEN (C$(N)))
170 C$(N) = C$ : GOSUB 3000
180  VTAB 23: HTAB 5: PRINT "AGAIN <Y/N>";: INPUT Y$ : IF Y$ < > "N"
     THEN 130
190  GOSUB 2000
200  GOSUB 3000: VTAB 20: PRINT "OK!"
900  END
1000 I = 43139:T = 0
1020 T = T+1:C$(T) = ""
1030 I = I+1:S = PEEK (I):C$(T) = C$(T)+CHR$ (S-INT (S / 128) * 128)
1040  IF S < 128 THEN 1030
1045  PRINT "<";T;">"; SPC ( 3-LEN ( STR $ (T)));C$(T);SPC(14-LEN(C$(T)));
     : IF T / 2 = INT (T / 2) THEN PRINT
1050  IF T < > 28 THEN 1020
1060 L = 0 : RETURN
2000 I = 43140
2010  FOR T = 1 TO 28: FOR K = 1 TO LEN (C$(T))-1
2020  POKE I,ASC ( MID$ (C$(T),K,1)):I = I+1
```

```
2030  NEXT K : POKE I,ASC ( RIGHT $ (C $ (T),1)) + 128:I = I + 1
2040  NEXT T
2050  RETURN
3000  TEXT : HOME : VTAB 4: FOR T = 1 TO 28
3010  PRINT "<";T;">"; SPC ( 3 - LEN ( STR $ (T)));C $ (T); SPC ( 14 - LEN (C $ (T)));:
      IF T / 2 = INT (T / 2) THEN PRINT
3020  NEXT T
3030  RETURN
```

程序的用法如下:

```
]RUN
      <1>  INIT              <2>  LOAD
      <3>  SAVE              <4>  RUN
      <5>  CHAIN             <6>  DELETE
      <7>  LOCK              <8>  UNLOCK
      <9>  CLOSE             <10> READ
      <11> EXEC              <12> WRITE
      <13> POSITION          <14> OPEN
      <15> APPEND            <16> RENAME
      <17> CATALOG           <18> MON
      <19> NOMON             <20> PR#
      <21> IN#               <22> MAXFILES
      <23> FP                <24> INT
      <25> BSAVE             <26> BLOAD
      <27> BRUN              <28> VERIFY
        NO.<1...28>?17
      NO.17:                     -->?TAI
      <1>  INIT              <2>  LOAD
      <3>  SAVE              <4>  RUN
      <5>  CHAIN             <6>  DELETE
      <7>  LOCK              <8>  UNLOCK
      <9>  CLOSE             <10> READ
      <11> EXEC              <12> WRITE
      <13> POSITION          <14> OPEN
      <15> APPEND            <16> RENAME
      <17> TAI               <18> MON
      <19> NOMON             <20> PR#
      <21> IN#               <22> MAXFILES
      <23> FP                <24> INT
      <25> BSAVE             <26> BLOAD
      <27> BRUN              <28> VERIFY
   AGAIN <Y/N>?N $
```

190. 改变 DOS 磁盘格式参数的加密方法

标准的 DOS 磁盘在进行初始化工作时就完成了规定的磁道格式及参数的设置。这一工作的完成是由 DOS 系统程序中的初始设定子程序($ BC56～ $ BCC3), WRITE 子程

序($B82A～$B8B7)，READ 子程序($B8DC～$B943)和 RDADR 子程序($B944～$B99F)等程序完成的。有关 DOS 磁盘格式以及参数形式是由这些程序中的相应参数决定。改变这些参数值可以制作出非标准格式的磁盘。如同步字节值，住址区起始标记，结束标记等参数的改变都能使正常的拷贝程序无法成功的复制，而达到保护的目的。磁盘格式主要参数住址表如下：

表 190-1　磁盘格式主要参数位址表

标准参数值			WR 子程序	RE 子程序	RD 子程序	初设子程序
位址区	起始标识符	$D5 $AA $96			$B955 $B95F $B96A	$BC7A $BC7F $BC84
	结束标识符	$DE $AA $EB			$B991 $B998	$BCAC $BCB3 $BCB8
资料区	起始标识符	$D5 $AA $AB	$B853 $B858 $B85D	$B8E7 $B8F1 $B8FC		
	结束标识符	$DE $AA $EB	$B89C $B8A3 $B8A8	$B935 $B93F		

例如将住址栏和资料栏的结束标记 $DE／$AA／$EB 中的 $AA 改变为 $92，可做如下操作：

```
CALL-151
*B8A3:92
*B99B:92
*B93F:92
*BCB3:92
*FP
INIT HELLO
```

这样格式化后的磁盘可成为非标准格式磁盘。对于 DOS 磁盘格式有关参数，其取值不是任意的，$D5，$AA，是磁盘数据的保留字节，不允许在位址区内或资料区出现，以确保某些标志的“唯一性”免受破坏，造成磁盘数据的混乱，以至完全不能使用。

191. 怎样改变DOS命令的入口地址

在DOS系统中，＄9D1E～＄9D55内存区间内按照DOS命令表的标准顺序，存放DOS命令处理程序的入口地址表，这些入口参数改变必然引起命令功能改变，例如为了屏蔽CATALOG或LIST命令的功能，可将其相对应入口处变为RUN命令的入口(＄A4D1)，那么在打入CATALOG或是LIST命令时执行的是RUN命令功能，原来功能失效而达到防止列程序清单或目录的目的。

192. 怎样移动VTOC表保护磁盘

标准DOS的磁盘文件管理主要是依靠VTOC(Volume Table Of Contents)表提供磁盘扇区使用的状况来管理磁盘。正常的DOS将VTOC表固定放于第17磁道0扇区。移动VTOC表后，正常DOS将因找不到VTOC表而不能正常读写磁盘。

移动VTOC表的做法可在BASIC状态或监控状态下，将VTOC表所在磁道号送入有关地址:

POKE 44703, N

POKE 44764, N

POKE 44033, N

INIT HELLO

其中N表示VTOC表所在的磁道号，取值范围是3＜N＜35的整数。

把正常DOS系统磁盘文件转到移动过VTOC表的磁盘上，在读和写操作时相应改变内存44033地址内为VTOC表所在磁道号。

对于移动过VTOC表的磁盘，只要＄11道未使用，还可以复制一个假的目录磁道在＄11道上，在真的VTOC表中指出＄11已被占用，对这样的磁盘用CATALOG命令列目录时，看到的是假目录。此种保护磁盘的方法是双目录保护法，如果把两个目录用不同磁盘格式分配给两个区域使用，各自的VTOC表同时注册两个区的磁道扇区使用状态，这就是双DOS保护法。这是一种行之有效的保护方法，因为往往一种解密方法不会对两种格式都有效。

193. 怎样使用备用磁道保护磁盘

标准的DOS系统使用的磁盘磁道是从0道至35道，但实际上可以使用更多的磁道，使磁盘增4个磁道。从磁盘的第36道开始称为备用磁道。如果将某些信息资料存放在备用磁道上，而正常的拷贝程序难以复制，因而起到保护磁盘的作用。

在BASIC状态下键入下列内容:

POKE 44703,36

POKE 44741,74

POKE 44764,36

POKE 44033,36

POKE 48894,37

POKE 44725,148

然后键入引导程序(HELLO)

INIT HELLO

这样初始化生成的新盘已将 VTOC 表及目录扇区放置于备用磁道 36 道上。

194. 如何制作 1／2 磁道的加密盘

有些软件盘的加密方法是把标准的 DOS 系统进行修改，使它变为 1／2 磁道存放信息，而标准磁道不存放信息。这样在使用拷贝程序进行复制时，就无法正确地读写信息，从而达到加密的目的。

1／2 磁道加密磁盘的方法如下:

1. 引导 DOS 系统。
2. 插入要加密的磁盘。
3. 键入下列信息

```
*
BCDF- 86
BCE0- 2B 85 2A C9 0C 90 02 E6
BCE8- 2A AD 78 04 C9 0C 90 03
BCF0- EE 78 04 A5 2A 60
*
B9A0- 20 DF BC EA
*
BCDF    86 2B       STX    $2B
BCE1    85 2A       STA    $2A
BCE3    C9 0C       CMP    #$0C
BCE5    90 02       BCC    #BCE9
BCE7    E6 2A       INC    #2A
BCE9    AD 78 04    LDA    $0478
BCEC    C9 0C       CMP    #$0C
BCEE    90 03       BCC    $BCF3
BCF0    EE 78 04    INC    $0478
BCF3    A5 2A       LDA    $2A
BCF5    60          RTS
BCF6    00          BRK
*FP
```

4. 检查无误后，键入一个问候程序，如:

```
10 PRINT"CEC-I DOS 1/2 TRACK DISK"
]INIT HELLO
```

此盘初始化后，就与普通 DOS 盘一样使用，建立磁盘文件，进行读写操作。但是这张磁盘用一般的拷贝程序却无法复制。

195.　什么是“炸弹法”保密

“炸弹法”保密在国外计算机市场上较为普及。这种加密原理是：在软件编制过程中设计了“定时炸弹”(引爆计数器及爆炸措施)，当使用者免费拿回软件并使用了几次尝到了“甜头”后软件会发出危险信号。如需要继续使用，就必须向软件商付款，索回该软件的密码钥匙，然后把密码告诉计算机。如这密码与该软件原定密码一致，危险信号就解除，软件可以继续使用。否则软件就会自动“炸毁”，留在使用者手中的。仅是一张普通空盘。

排除“炸弹”的方法通常可以分二种，一种是永久性地全部排除“炸弹”，另一种是局部排除“炸弹”，这就要看使用者向软件商付款多少而确定继续使用的次数了。

下面的程序是一个简单摸拟“定时炸弹”法加密磁盘的例子。

```
5 REM PROGRAM 195.1
10 D$ = CHR$ (4)
20 PRINT D$;"OPEN TAIZD"
30 PRINT D$;"READ TAIZD"
40 INPUT A
50 PRINT D$;"CLOSE TAIZD"
100 IF A = 5 THEN 200
110 A = A+1
120 PRINT D$;"DELETE TAIZD"
130 PRINT D$;"OPEN TAIZD"
140 PRINT D$;"WRITE TAIZD"
150 PRINT A$        160 PRINT D$;"CLOSE TAIZD"
190 GOTO 300
200 HOME
210 INPUT "MI MA:";MA$
220 IF MA$ < > "TAI-PIAO" THEN 290
230 PRINT D$;"DELETE TAIZD"
240 PRINT D$;"DELETE HELLO"
260 PRINT D$;"RENAME HELLO1,HELLO"
270 PR#6
290 PRINT D$;"INIT HELLO"
300 DEL 0,300
```

程序说明如下：

1. 该程序作为问候程序(HELLO)

2. 名为 TAIZD 的 T 型文件内容为引爆计数器的值，应在软件编制时，先建立好 TAIZD，并设置好初值 A = 0。

3. 程序的第 100 句中“5”为自定的允许用户试用软件的次数。当用户使用密码满五次后，软件就会自动爆炸(INIT 初始化)。

4. 程序中 260 句中提到的文件 HELLO1 是一个标准的问候程序，当知道密码后，

软件将解除“定时炸弹”改用 HELLO1 作为问候程序。

总之，上述介绍的加密方法只是为了说明这种加密思想。设计者可以进行改进，例如：

A 修改爆炸的措施或替换“INIT”命令；

B HELLO 程序加密；

C 设置开启 T 型文件的陷井；

D 使 RESET 和 CTRL－C 键失效等。

196. CTRL-RESET 键的特殊用法

每当按下 CTRL-RESET 键，中华学习机就会强迫中断当前工作进入 RESET (重置) 过程。其中有一项操作是判断 $3F4 (1012)地址的内容是否等于 $3F3 (1011)地址内容与 $A5 (165)异或加的结果。如果不等则进行一次热启动工作；如果相等，则进入由 $3F2 (1010) 和 $3F3 (1011)所指向的汇编程序中运行。利用这一点，可以设计出许多特殊的功能，如分别做下面的修改，按下 CTRL-RESET 会有不同的作用。

3F2：66 D5 70　重新运行一次 BASIC 程序。

3F2：69 FF 5A　进入监控状态，相当于 CALL-151；

3F2：4B D6 73　清除 BASIC 程序，相当于 NEW；

3F4：00　　　　重新启动 DOS，相当于 PR#6；

以上内容转成十进制数亦可在 BASIC 程序中直接使用。

197. CTRL-C 键的特殊用法

一个完整的 BASIC 程序，如果不想让别人查看，首先可以使程序自己死循环，没有结束点，其次封锁 CTRL-RESET 键，避免强迫中断，作为进一步的完善，还应该封锁 CTRL-C 的中断功能，否则将前功尽弃。在 APPLESOFT 中，CTRL-C 被认为是一种错误形式，并赋予错误代码 255，根据这个特性，可以在你设计的程序中把 CTRL-C 分离出来，单独处理。例如在程序中加入下列语句：

```
0     ONERR   GOTO   63998
   .
   .
   .
程序主体


63998 IF PEEK(222)=255 THEN 63998
63999 PRINT PEEK(222):GOTO 63999
```

运行程序时，按下 CTRL-C 后，就进入死循环。而遇到某种错误则反复输出错误代码。

198. 如何寻找文件名中的保密字符

在 DOS 系统中，有时文件名中会含有保密字符(控制字符)。对于含有保密字符的文件名，进行运行或装入等操作时，往往会出现下列信息:

FILF NOT FOUND (没有发现文件)

下面的程序运行后，可以帮助你找到文件名中的保密字符。程序如下:

```
5   REM PROGRAM 198.1
10  DATA 201,141,240,21,201,136
20  DATA 240,17,201,128,144,13
30  DATA 201,160,176,9,72,132
40  DATA 53,56,233,64,76,249
50  DATA 253 76,240,253
60  FOR I = 768 TO 768 + 27
70  READ V: POKE I,V: NEXT I
80  POKE 54,0: POKE 55,3
90  CALL 1002
100 END
```

运行后，可以看到出现系统提示符"]"而未有任何现象。这时，键入 CATALOG 命令后，便可得到含闪动的控制字符的文件名目录。如果想恢复正常的显示方式，可键入 PR#0。

199. DOS 操作系统下怎样拷贝文件

有时，需要把某张磁盘中的某个文件复制到另外磁盘上。如果要复制一个 BASIC 文件，方法很简单。只需要先用 LOAD 命令装入内存，而后再用 SAVE 命令存入另一个指定的磁盘中即可。但是，这种复制方法对于文本文件或机器语言文件，则比较麻烦。

下面介绍一种复制文件的方法，这种方法可适用于任何类型文件的复制工作。

在 DOS3.3 系统主盘中，有一个"FID"文件，是用机器语言写的程序。此程序共有 9 个功能模块。用户可将系统磁盘插入驱动器内，键入命令:

BRUN FID

屏幕上以菜单形式显示内容:

```
CHOOSE ONE OF THE FOLLOWING OPTIONS
    <1>  COPY FILES
    <2>  CATALOG
    <3>  SPACE ON DISK
    <4>  UNLOCK FILES
```

＜5＞ LOCK FILES

＜6＞ DELETE FILES

＜7＞ RESET SLOT &DRIVE

＜8＞ VERIFY FILES

＜9＞ QUIT

WHICH WOULD YOU LIKE?

这时，键入数字 1 且按下回车键。屏幕显示：

COPY FILES

SOURCE SLOT?(询问原盘插槽号)

DRIVE?(询问原盘驱动器号)

DESTINATION SLOT?(询问复制盘的驱动器号)

DRIVE?(询问要复制盘的驱动器号)

FILE NAME?(询问要复制盘的驱动器号)

用户可以根据实际的外设接口，来一一回答以上问题。

例如：用户使用双驱动器(1 和 2)，驱动器卡插第六号槽。

屏幕显示	键入
SOURCE SLOT?	6
DRIVE?	1
DESTINATION SLOT?	6
DRIVE?	2
FILENAME? 文件名	

此时屏幕出现提示：

INSERT DISK PRESS ＜ESC＞ TO RETURN TO MAIN MENU OR ANY OTHER KEY TO BEGIN

提示用户插入磁盘，或者按 ESC 键返回主菜单，或者按下其他健开始复制。

可按下 RETURN 键，便开始复制。当屏幕再次显示：

FILES 文件名

DONE

PRESS ANY KEY TO CONTINUE

此时，表示已复制完毕。

这种单一文件的复制方法，类似于 CP／M 操作系统中的 PIP 指令，无需考虑文件类型及文件在内存的位置，是一个非常实用的方法。

200. 如何拷贝任意磁道数的磁盘

有些软件是经过加密的，常见的方法是把某些信息写在备用磁道上，这样用普通的 COPY 程序是无法复制备份的。下面的程序可以对任意系统的软件进行拷贝，而且磁道数可以是任意的，即可拷贝常规 35 道盘，又可拷贝多于 35 道的加密软件盘。在使用本程序前，对目的盘进行需要磁道数的格式化。

为了加快运行速度，本程序的主要部分用机器语言编写，取名为"COPY-RM"，其中$C00-$C25是核心部分，$B40-$BFF是主程序，$C28-$C55是显示代码。

```
0B40- A9 05 85 FB 85 1C A9 06
0B48- 85 FD A9 A8 85 FA 85 FC
0B50- A9 D2 85 07 A9 D7 85 09
0B58- A9 00 85 1A A5 1B C9 01
0B60- D0 0E A0 16 B9 28 0C 99
0B68- 00 07 88 10 F7 20 ED 0B
0B70- 20 7F 0B 18 A5 1A 69 08
0B78- 85 1A C6 1C D0 DF 60 A9
0B80- 01 8D 0C 0C 8D 16 0C A9
0B88- 00 85 14 20 B0 0B A5 1B
0B90- C9 01 D0 0E A0 16 B9 52
0B98- 0C 99 00 07 88 10 F7 20
0BA0- ED 0B A9 02 8D 16 0C A5
0BA8- 1B 8D 0C 0C A9 02 85 14
0BB0- A9 10 85 08 A5 1A 85 06
0BB8- A9 07 85 11 A6 14 B5 07
0BC0- 81 FA F6 FA A5 06 8D 0E
0BC8- 0C A9 0F 85 15 A5 1A 8D
0BD0- 0F 0C A5 08 8D 13 0C 20
0BD8- 00 0C E6 08 C6 15 10 ED
```

```
0BE0- E6 06 A5 19 C5 06 F0 04
0BE8- C6 11 10 D0 60 AD 10 C0
OBF0- AD 00 C0 10 FB A0 18 A9
0BF8- A0 99 00 07 88 10 FA 60
0C00- A9 0C A0 0A 20 D9 03 60
0C08- 00 3A 01 60 01 00 11 00
0C10- 20 0C 00 20 00 00 01 00
0C18- 00 60 01 20 20 20 20 20
0C20- 00 01 EF D8 60 00 85 9E
0C28- 49 4E 53 45 52 54 60 53
0C30- 4F 55 52 43 45 60 44 49
0C38- 53 4B 60 7C 43 52 7E 49
0C40- 4E 53 45 52 54 60 54 41
0C48- 52 47 45 54 60 44 49 53
0C50- 4B 60 7C 43 52 7E FF FF
```

此机器语言再由BASIC程序调用，这是为了程序简单。运行开始，先键入需要的磁盘的磁道数，程序将磁道数控制在35至40之间，然后再键入驱动器的个数，回车即可。

```
5 REM PROGRAM 200.1
100 PRINT CHR$ (4);"BLOAD COPY-RW"
110 B$ = "0123456789ABCDEF0123456789ABCDEF01234567"
120 HOME : PRINT TAB( 12);" * * * COPY DISK * * * ": PRINT
130 INPUT "TRACK NUMBER ?";T
140 IF T < 35 OR T > 40 THEN 130
150 POKE 25,T: INPUT "DRIVE NUMBER ?";D
160 IF D < 1 OR D > 2 THEN 150
170 POKE 27,D: IF D = 1 THEN PRINT : PRINT : GOTO 200
180 PRINT "SOURCE DISK INSERT IN D#1"
190 PRITN "TARGET DISK INSERT IN D#2"
200 INPUT "<RETURN> TO BEGIN...";A$
210 VTAB 20: PRINT LEFT$ (B$,T)
```

```
220  CALL 2880
230  VTAB 20: HTAB 1: PRINT "THE END !"
240  DEL 100,100
```

201. 什么是文件脱解法

把某些程序从加密盘上转移到标准 DOS 系统磁盘上的过程叫做文件脱解。进行文件脱解的基本原理是：把加密磁盘文件装入内存，将其移动到不受 BOOT 启动磁盘影响的区域，然后正常启动 DOS 系统，再把程序重置，最后存入正常的 DOS 系统磁盘上。通常启动一张无程序的 DOS3.3 软磁盘时，内存 $ 9000－$ 95FF 地址区域是不受影响的，这是隐藏程序的安全区域。

下面以 BASIC 程序和二进制文件的脱解为例，介绍有关文件脱解的办法和技巧。

一． APPLESOFT BASIC 程序由加密系统脱解

1． 首先制作一个 DOS3.3 系统空磁盘，要求 HELLO 引导程序是空白的或是及短小的程序。具体作法：

(1) 启动 DOS3.3 系统磁盘；

(2) NEW 清除内存；

(3) 换上空白磁盘，键入 INIT HELLO

2． 启动加密磁盘系统

(1) PR#6

(2) LOAD 文件名装入要脱解的 BASIC 程序。

3． CALL－151 进入监控状态；

(1) 查找 $ 67， $ 68， $ AF， $ B0 地址，记下程序装入的起始地址，结束地址，并计算出其长度。

(2) 把程序移至"安全区域"

例如，原程序起始地址为 $ 800，结束地址为 $ 3000，则做如下操作：

＊6000＜800.3000M

程序即移至内存 $ 6000 地址开始的区域。

4． 启动标准的 DOS 3.3 空磁盘

＊C600G （将正常 DOS 系统装入内存）

5． 重置程序及有关指针

＊800＜6000.8800M

将 $ 67， $ 68 及 $ AF， $ B0 地址内容置为程序起始和终止。

6. CTRL－B 回到 APPLESOFT 状态；

7. SAVE 文件名将程序存入正常的 DOS 3.3 系统磁盘上，完成 BASIC 程序的脱解。

二． 二进制文件由加密系统脱解

1. 首先制做 DOS 3.3 系统空磁盘；

2. 启动加密磁盘系统；

3. BLOAD 文件名上装入要脱解的二进制文件。

4. CALL−151 进入监控;

5. 查询＄AA72, ＄AA73 及＄AA60, ＄AA61 地址内容，找出程序起始地址及程序长度;

6. 计算移动程序前后结束地址;

7. 移动程序到安全地址区域;

8. 启动 DOS3.3 空磁盘;

9. 把移动过的程序重置;

10. 更改＄AA60、＄AA61、＄AA72、＄AA73 地址内容为程序的长度及程序的起始地址;

11. 回到 BASIC 状态，将二进制文件用 BSAVE 命令存盘，即完成二进制文件的脱解。

202. 怎样使用 CRAZY COPY 软件

CRAZY COPY 软件是几个流行的 COPY 工具软件之一，它使用方便，操作简单。下面介绍使用方法:

CRAZY COPY 软件专用于复制那些磁盘上用常规的方法难以复制的磁盘。利用一次一位元的方式，几乎可以复制所有加密的磁盘。

它的复制速度比 LOCKSMITH 或 BACKT−UP II 快 2 至 3 倍，并且不必修改参数也可以复制。对 1/2 磁道，允许 0.5 磁道的增量。可以复制超出通常磁道数范围外的磁盘。另外，还提供磁道影象表，便于初学者使用。

由于本软件对硬件敏感，故而使用时要使驱动器的转速误差尽可能调整为零。若用双驱动器的话，则要让二者同步。另外，要避免复制很难启动的磁盘。

使用开始时通过主机 6 号槽启动本软件。几秒种后屏幕显示主功能表。

屏幕上自上而下出现磁道状态，版权资料，接着问原盘所在驱动器号，回答 1 或 2(若直按回车键，则采用软件自定值 1)。

其次输入复制盘所在驱动器号，回答 1 或 2(自定值是 2)。

随后问结束磁道增量(自定值是 ＄22)半磁道时亦可自由设定。

最后输入磁道增量(自定值是 ＄1)，半磁道时可设为 0.5 或 1.5。

如果使用双台驱动器时，把原盘与复制盘分别放入 1、2 号驱动器，对磁道设定及增量没有特别要求的话，那么直接按回车键即可开始复制了。

如果在回答上述五个问题时答错了，那么按 ESC 键可重新回答问题。

屏幕上显示的磁道状态表告诉我们每一磁道(半磁道在下半部)的复制状态，其中的记号含义如下:

R————读取

W————写入

∧————完成

* ————空磁道或资料错误

203. 怎样使用快速拷贝 DISK MUNCHER 软件

快速拷贝 DISK MUNCHER 软件也是目前较为流行的一种工具软件，它的最大特点是拷贝速度快,一分钟左右即可复制一张盘，拷贝的质量也较高。另一个特点是操作非常简便。DISK MUNCHER 软件的版本号很多，我们以 1.0 版本为例，说明使用方法。

首先把 DISK MUNCHER 软件盘插入 1 号驱动器内，开机启动，这时屏幕显示如下：

```
DISK MUNCHER 1.0 - COPYRIGHT JUNE 1983
WRITTER BY THE STACK - CORRUPT COMPUTING
* * * * * * * * * * * * * * * * * * * * *
*      [1] CATALOG SOURCE              *
*      [2] CATALOG TARGET              *
*      [3] COPY DISK                   *
*      [4] BOOT DISK                   *
*                                      *
* * * * * * * * * * * * * * * * * * * * *
```

然后，把原盘与空白盘分别插入 1 和 2 号驱动器内。

键入 1，表示查看原盘的目录及磁盘内部扇区的使用情况。

键入 2，表示查看复制盘的目录及磁盘内部的使用情况。

键入 3，表示要进磁盘拷贝。这时，查看磁盘是否按规定放入指定驱动器内，按下回车键拷贝开始。

键入 4，表示要退出拷贝状态。键入 Y，即可启动磁盘。

当进行上述任何一项操作时，按下 ESC 键，即可返回主菜单。

七、汉字系统

204. 如何列出主机内汉字字库

中华学习机主机内存有汉字系统，可以在程序中很方便地使用汉字，主机内汉字字库分为 1 到 87 区，其中 10 至 15 区为空白。下面的一行程序可以方便地将字库中的所有汉字在屏幕上显示：

```
10 DEF FN A(X)—(x>15)+(x>15)+(x>27)+X+28: HOME: PRINT CHR $ (18): FOR I=1 TO
87: A=FN A(I): PRINT: PRINT I“区:”; I+6*(I=9): FOR J=1 TO 94: PRINT CHR
(127) CHR $ (A) CHR $ (FN A(J));: FOR K=0 TO 1: B=PEEK(49152): POKE 49168, 0:
C=B=155: I=B*C+I: J=B*C+J: B=B>127: D=B+D−2*B*D: K=C−D+1: NEXT K, J, I
```

输入程序时为了使程序行的字符限制在 255 个之内，所有的“PRINT”定义符用“?”代替，并省略所有的空格。

使用时键入 RUN 命令运行，按下任意键可以暂停显示，再接任意键继续，用 ESC 键结束，这样就可以在屏幕上显示主机内字库中的所有汉字了。

205. 怎样使用 STC 2.0 汉字系统

STC 2.0 软汉字系统是我国近几年来在 APPLE Ⅱ微机及其兼容机上开发的较为优秀的汉字系统软件。由于它操作简便，功能较强，已被广泛地应用。中华学习机在制造时已装入硬汉字系统，但是，也可以使用一些优秀的软汉字系统以开发更多的功能。下面简单介绍 STC 2.0 的使用方法。

一、启动系统

STC 2.0 汉字系统包括两张磁盘，一张称为 STC 2.0 系统盘；另一张称为字库盘。将系统盘插入 1 号驱动器内，开机屏幕上便会出现“STC 软汉字系统”等字样，按任何键后左上角出现 APPLESOFT 提示符“]”。这时，STC 系统装入了主机内存，用户可以自由地编写和执行夹有汉字的程序。

二、汉字的输入方法

如果在程序设计中，输入汉字时，则必须放入字库盘(2 号驱动器)。如果使用单台驱动器，则通过下列命令，来改变字库所在的驱动器。

STC Sm　(m 表示字库盘所在驱动器的槽号，自定义为 6)

STC Dn　(n 表示字库盘所在驱动器号，自定义为 2)

字库盘放后，键入 CTRL−L，主机发出“嘟”声，屏幕左下角出现“?”，表示已进入汉字输入准备状态。

汉字输入有下列两种方式：

1. 国标码输入

国标码由四个数字组成，使用的区域在1601到6080之间。键入四个数字，屏幕上就出现该国标码所对应的汉字，与此同时，此汉字就被送入内存。

如:? 1601　　啊

2. 拼音输入

输入拼音的前四个字母(不足四个字母时，按下空格键)。屏幕就会显示出八个同音汉字供用户查找，找到后键入汉字的编号0～7。

如果出现的汉字不是所需要的，可按下空格键向后查找，按下←键向前查找。

三、造字及改字功能

一级字库内容为1601到5589，每区后六个字不用(如：18区中的1895～1900)。STC所能接受的国标信息为1601至6080，容量大于一级字库，多余部分供用户存放自己造的字或图案。

键入CTRL-W，系统进入造(改)字状态，然后请输入造(改)字的国标码，这时屏幕右方出现15＊16的线框，框内有该国标码对应的内容，使用A、Z、←、→四个键控制光标上、下左、右移动，用空格键决定光标所在位置是否画点。

如果对造(改)的字不满意，按下ESC便退出造(改)字状态。

如果修改满意，则按下回车键，修改的字就被写入内存。屏幕下方询问是否要把改好的字存入磁盘。请用Y(是)或N(否)来回答。

四、屏幕显示纵横向放大

有时，需要将屏幕上的汉字或字符进行放大显示。STC系统提供下列命令:

STC Rm　　放大m倍

STC R0　　恢复原状;

POKE 4934，n　　纵向单独放大n倍;

POKE 802，n　　横向单独放大n倍;

STC系统还设置了STC A命令，让用户方便地调整ASCII字符在屏幕上的纵向位置，以便和相应的汉字显示配合。命令为:

STC An　　n为0至8的整数。

五、打印机的操作

STC 系统能够方便地连打印机输出打印汉字。命令如下:

STC B1　打开打印机，设置平常打印方式;

STC B2　打开打印机，设置倍密度打印方式;

STC B0　关闭打印机;

STC Vn　n为纵向行距，当n取8时，即通常行距;

STC Ln　n为打印机每行打印的ASCII字符数。一个汉字宽为两个ASCII字符数。一个汉字宽为两个ASCII字符。

屏幕硬拷贝的方法如下:

PRINT CHR $ (4); “PR#1”: POKE 1913，1: PRINT CHR $ (17): CALL 6721

六、STC系统上几个程序介绍

ASCII.EDITOR是一个供用户自造ASCII字符的工具程序。ASCII.EDITOR.LIB为它的小字库。

ASC 是 B 型文件，STC 系统已造好一套字形宽的 ASCII 字符。如果用户运行程序前先将该文件装入内存(BLOAD ASC)，即可用这套字符。否则仍使用通常的 ASCII 码字符。

CHARS 是一个 B 型文件，执行它可以打印整张字库盘中的汉字。

INIT 是一个 B 型文件，执行它可对新盘进行格式化，并自动装入 STC 系统和快速 DOS。

206. STC 4.0 汉字系统的主要功能是什么

STC 4.0 汉字系统是在 STC 1.0 与 STC 2.0 基础上作了较大修改和扩充后的新版本，需要含有 64K RAM 的 APPLE Ⅱ微机及中华学习机硬件的支持。作为汉字系统，新版本着力于加强在汉字输入和输出方面的功能，但仍保存与原 STC 2.0 版本的兼容。STC 4.0 版本的主要功能特点如下：

一、多种汉字编码输入，输入速度大大提高

STC 4.0 汉字系统采用了通用的 3 至 5 倍编码转换的方案，大大加强了在微机上实现多种汉字编码的适应性，任何码长在 3～5 位的汉字编码方案极容易实现。目前 STC 4.0 系统已在原编码(拼音、区位)基础上增加了表形码(由法籍华人陈爱文先生提出的一种具有全新编码理论的码)；钱码(由中国中文信息处理学会会长钱伟长教授提出的编码)；以及笔形、电报、数形、仓颉等编码。其次，STC 4.0 系统中的各种编码系统都可以直接从一、二级汉字字库中进行检字。

为了进一步提高汉字输入的速度，在 STC 4.0 系统中还提供了词码输入汉字串，设置驱动器常开等方式。

二、多种屏幕显示方式和打印方式，汉字输出形式丰富多采

1. 屏幕显示方式

屏幕显示汉字或字符的大小均可分别在纵横两个方向自由放大，显示位置可任意设定，显示方式可选择正向、反向等多种，并有多套字符可供选择，使屏幕显示更加丰富多彩。

2. 打印方式

可由用户方便地选择打印机槽号，打印宽度，纵横向的放大或压缩、旋转、调整 ASCII 字符与汉字的边沿等。打印方式多达 18，000 余种，在这些参数选择时，可使用设置与添加两种方式，大大方便了用户的操作作用。

三、其它扩充功能

1. 拼拆方式造字功能

为了使用户方便地编辑汉字字形成图形，STC 4.0 系统设有汉字复盖功能，用于任意拼装汉字部首，偏旁，也可以用于各种图形符号的组合式分解。

2. 小字库编辑管理功能

STC 4.0 系统可以对小字库中汉字进行选择删除，亦可部分或全部列出小字库汉字清单，并设有显示暂停退出功能。

3. 增加的 DOS 管理命令

新增加的 TYPE 命令，用于查询磁盘上各种文件的文本内容，更改 CATALOG 命令执行的功能，使 CATALOG 命令在执行中随时可用 ESC 键中断。

4. 子系统生成功能

对某些有用盘片不需要造字、打印、多种编码输入等功能，可以另外生成子系统盘。这样对节省内存和外存，提高运行速度都有利。

5. 采用盘符来区别系统、字库、编码盘

STC 4.0 系统包括有系统盘，编码盘、一级字库盘、二级字库盘，为了防止用错盘，各盘都有各自的盘符，系统会自动判别、提示。

四、系统主盘上的几个外围文件

1. HELLO

系统主盘上的引导程序，除显示软件名称、制作者外，兼有 SETUP 功能，可让用户在进入系统前预先设定各类外围设备槽号和驱动器号，以及是否使用 128K RAM 卡，是否需要造字等。

2. PRINT 16K

运行该程序后，在子系统盘上即可使用 STC 4.0 打印功能。

3. CHARS

打印一级、二级字库盘程序。

4. SET 128K

用户可以使用 128K RAM 卡，以扩充用户内存空间。

5. ASCII EDIT

用户可通过该程序自由编辑 ASCII 字符集。

6. INIT 4.0 SLAVE

生成 STC 4.0 子系统盘。

7. CWDLIB

内存小字库和缩写键进行自由的存取、调入、列目录及拼接等管理。

8. SET 1～SET 20

系统提供 20 套 ASCII 字符以供用户使用，可用 BLOAD 命令调入。

207. 在中华学习机上如何使用 STC 软汉字系统的 BASIC 程序

目前广泛使用的 APPLE Ⅱ 微机上许多的软件是由 STC 汉字系统支持的，为使中华学习机上能够运行 STC 汉字系统下编制的 BASIC 程序，用下面的方法可以达到移植的作用。

汉字在 BASIC 程序中是以字符串的形式出现的，其代码不同于区位码，不同的操作系统下其代码是不同的，所以中华学习机的硬汉字系统和 STC 软汉字系统其汉字的代码是不同的，一般是不能通用的。

STC 软汉字系统的汉字代码是以＄1B 为首加两字节国标区位码构成，对一些特殊的字符由区位码修正：＄3A(冒号：)为＄64，＄2C(逗号，)为＄65，＄22(引号)为＄66，＄00(行结束标志)为＄67，＄0D(回车)为＄68。

中华学习机汉字代码是以＄7F为首加两字节区位修正码构成，修正的规律是将区位码先加上＄1C，若结果大于等于＄22则再加1，若又大于等于＄2C再加1，若又大于等于＄3A再加1。

STC软汉字系统下的BASIC语言程序在中华学习机上运行的方法如下：首先输入下面的机器语言程序：

```
9000-A5 68 85 09 A5 67 85 08
9008-A0 00 A5 08 85 06 A5 09
9010-85 -07 B1 06 85 08 C8 B1
9018-06 85 09 F0 2D A9 A3 20
9020-ED FD C8 B1 06 AA C8 B1
9028-06 20 24 ED A0 03 C8 B1
9030-06 F0 D5 C9 1B D0 F7 A9
9038-7F 91 06 C8 B1 06 20 4B
9040-90· C8 B1 06 20 4B 90 4C
9048-2E 90 60 C9 5F 30 18 C9
9050-64 30 09 38 E9 64 AA BD
9058-7E 90 D0 08 88 A9 38 91
9060-06 C8 A9 38 91 06 60 18
9068-69 1C C9 22 30 F6 69 00
9070-C9 2C 30 F0 69 00 C9 3A
9078-30 EA 69 00 D0 E6 59 4B
9080-41 00 2A
```

然后把要运行的BASIC程序装入内存，CALL 36864运行上面的机器语言程序。转换后的程序中如果有STC汉字系统的特殊功能语句，应删除或修改。

208. 怎样使用中华超级汉字系统

中华超级汉字系统由吴新和朱炜两位同学合作研制，在APPLE Ⅱ上开发的汉字系统。本系统适用于配有容量64K RAM以上的APPLE Ⅱ机及中华学习机。中华超级汉字系统使用方法如下：

一、汉字输入

启动系统盘后，键入CTRL-RESET键，把字库盘插入2号驱动器内。屏幕左下角出现提示符"]-ASCII"，表示汉字输入已进入准备状态。汉字的输入方式除常用的"区位"和"拼音"输入法外，加有7种编码可供用户自动定义。

1. 拼音输入法

按ESC键，再按CTRL-B，此时屏幕提示出"编码"两字。键入2，按下CTRL-L键，此时屏幕出现提示符为"拼音"两字。即可进行汉字拼音输入检字。

拼音输入时，某些声母或韵母的代用码与一般IBM中文输入系统相同。即：

SH→U CH→I ZH→A AI→L AN→J ANG→H AO→K EN→F

ENG→G ING→Y ONG→S ü→V

2. 区位码输入法

按ESC键；再按CTRL-B键，此时屏幕提示为"编码"两字。键入L，按下CTRL-L键，此时屏幕出现提示符为"区位"两字，即可进入区位码检字。

3. 用拼音输入法查找区位码

用拼音找出"中国"两字，按CTRL-L退出汉字输入命令：LST↓即可得到上述两字的区位码。

4. 程序中如何用区位码调用汉字

当用户已知汉字的区位码，则在程序中可通过区位码直接调用汉字。命令格式如下：
PRINT CHR $ (126)；CHR $ (31 * 区位码高位)；CHR $ (31 * 区位码低位)

5. 字库盘所处位置定义及单驱动器用户的使用方法

命令格式：&DIM Ss1，Dd1，Ss2，Dd2

其中 s1，s2 均为槽号；d1，d2 分别表示一、二级字库盘所在的驱动器号。

单台驱动器用户使用方法：启动系统后，键入：

&DIM S6，D1，S6，D1 ↓

6. 双驱动器同时使用一、二级字库方法：

进入系统后，把一、二级字库盘分别插入 D1，D2 驱动器内，键入：

&DIM S6，D1，S6，D2 ↓

即可。

二、屏幕显示功能

1. 编辑

与 APPLESOFT 相同。

2. 功能键命令

(1) 字体类

键入 CTRL-A，再键入参数 0～9。将 ASCII 字符体移行，形成有行差的字体八种，参数 9 是将字体拉长。运行下列程序：

```
10  FOR K=0 TO 9
20  PRINT CHR $ (1); K; " TAI" ;
30  NEXT K
```

键入 CTRL-F，再键入参数(A 或 C)(0～3)，可使用字体进行旋转，参数 A 表示 ASCII 字符，参数 C 表示汉字；0～3 分别代表顺时针旋转 0，$\pi/2$，π，$3\pi/2$。例如运行下列程序：

```
10  PRINT  "TAI PIAO"
20  PRINT  CHR $ (6); " A3"; "TAI PIAO"
30  PRINT  CHR $ (6); "C3"; "中国"
40  PRINT  CHR $ (6); "C1"; "中国"
```

键入 CTRL-K，不加参数。作用是把所有大写字母按小写字母输出。CTRL-V 可退出 CTRL-K 命令。

键入 CTRL-R，再键入参数 0 至 3。作用是选择内存中的四种 ASCII 点阵。这些点阵在启动时定义成标准 ASCII 集，用户可根据自己需要定义系统盘中的多种花体，这样显示的就是花体 ASCII。

(2) 屏幕类

CTRL-P：清屏。

CTRL-J：光标下移一行。

CTRL-T：消除"ASCII"字样，反之亦然。

CTRL-O：保存当前光标位置。

CTRL-Y：恢复光标。

3. &命令

&END	退出中文系统;
&RUN	进入中文系统;
&CLEAR	清内存字库;
&TOX，Y	字符点定位一页;
&HGR	设置高分辨率第一页;
&HGR2	设置高分辨率第二页;
&DIM Ss1，Dd1，Ss2，Dd2	定义字库盘。

三、打印功能

要使用打印机前，必须装入打印程序模块。即：LOAD PRINTER.SUS ↓。用户再使用下列操作命令：

CLTP Aa，Bb，Ll，Rr，Ss

其中：

Ad 是打印字体选择，共三种。A0 是标准字体打印；A1 是纵向压缩字体；A2 是纵向扩大字体。

Bb 是设置密度打印方式。B1 是标准密度打印；B2 是倍密度打印。

Ll 是设置打印行宽。一般 80 列打印机可设 60，倍密度打印行宽可设 120。当 l≤8 时，关闭打印机。

Rr 是设置打印行距，通常为 r／72 英寸。

Ss 是设置打印机所在槽号，CLTP 命令参数的自定义值为：

CLTP A0，B1，L1，R4，S1

CTRL−Q 是设置打印字体横向扩大 1~8 倍，命令格式为：CTRL−Qn 其中 1≤n≤8。

当 n＝1 是标准字体。

CTRL−N 是设置或取消字体纵向扩大 1 倍。

四、小字库管理

在系统盘中有一个 MANAGE.AUX 程序：它是用户字库管理程序，共有六项功能：

1. 调入字库 2. 合并字库 3. 存字库 4. 列字库 5. 造字 6. 删字

字库管理程序全部采用汉字提示，可按提示操作，由于有两个字库(被操作字库和中文提示字库)同时被调入内存，势必涉及切换问题，因此请不要随意中止程序。

五、辅助程序

1. 128K 汉字快速检字

本系统可将 128K RAM 卡作为虚拟磁盘来存字库，从而使检字速率进一步提高。如果有两块 128K RAM 卡，则可把一、二级字库均可调入。使用方法如下：

键入：RUN RAM128KSET.AUX ↓

2. STC 转换程序

用户过去用 STC 开发的软件如加入本系统支持，可使用 STC 转换程序(STC EXCHANGE.AUX)，使用方法如下：

(1) 键入 BRUN STC EXCHANGE.AUX。按提示输入文件名。

(2) 用&RUN 命令进入汉字系统。

(3) 运行 STC EXCHANGE.AUX 程序便可使 STC 运行下的程序转入本系统运行。

3. 国标码输入法

为方便用户，本系统除拼音、区位两种输入法外，另设有国标码供选用。使用方法如下：

LOAD GUO.BIAO.AUX↓

编码号为 3。

209. 怎样使用超级汉字系统

超级汉字系统也是一种软汉字系统，该系统不用增加和更改任何硬件，仅使用两张软盘(一张系统盘、一张字库盘)，就可以在中华学习机上使用了，该系统软件与 DOS3.3 完全兼容，可以在 APPLESOFT 状态下，用拼音、区位、五笔字形等方法直接输入全部国标一级和二级汉字；显示出的汉字为 16＊16 仿宋体点阵，字型美观大方；系统还具有造字与改字功能；打印时有 16 种大小不同的字体，同时，还可根据用户的需要而选用 9 针或 24 针打印机，操作简便实用。

一、汉字输入

系统启动后，键入 CTRL−L 后，屏幕左下角出现"区位／拼音"。这时，可以输入区位或拼音。

使用←、→键进行检字；拼音输入时，不足四个字母时，使用空格键。

二、打印机

CTRL−P 是打印机连接开关，每按一次，与打印机的连状态改变一次。在程序中可使用：

POKE 814,1 开关打印机

POKE 814,0 关打印机

CTRL−Z 是打印字型转换开关，本系统连接 9 针打印机时，可打印出 16 种大小不同的字体，用此命令可以随时在最大和最小字体之间切换。在程序中可使用：

POKE 815,n 其中 n 表示打印字型，$0 \leqslant n \leqslant 15$

另外在打印时，还可以选择下列命令：

POKE 813,K 控制打印行距，$0 \leqslant K \leqslant 255$；

POKE 779,l 控制打印行宽，$60 \leqslant l \leqslant 240$。

三、其它功能

CTRL−A 改变字库驱动器的编号，开机后，系统自动定义字库盘使用 2 号驱动器。对于单驱动器的用户，可在系统调入主机后，取出系统盘，换上字库盘，然后按下此键，就可正常使用。

CTRL−W 造字功能键，用此键可立即进入造字状态，随后用户可以随意设计，编制自已定义的汉字、图形或符号。

CTRL−O 显示或打印用户小字库的内容。显示时，每个汉字前边还给出该字库中内

容。

CTRL-B 清除机内小字库中的所有内容。使用此键，可对小字库进行清理，更新，也可在特殊要求下，让程序与小字库分离后单独处理。

210. 在中华学习机上如何使用“五笔字型”输入法输入汉字

在中华学习上使用汉字，有多种输入方法，“五笔字型”汉字输入方法要求有两张磁盘，其中一张是“中华学习机五笔字型汉字系统盘”，另一张是“中华学习机五笔字型汉字字库盘”。

五笔字型汉字系统盘可以采用开机启动或键入 PR#6 命令的方法启动，启动后屏幕上出现有关五笔字型输入的画面，并有“嘟嘟”两声，呈现提示符“]”和光标。表示系统盘启动成功。如果使用的是两台驱动器，就把字库盘插入 2 号驱动器；如果使用的是一台驱动器，就把系统盘撤出，放入字库盘按下 CTRL-A 键。然后就可以在程序中或屏幕上输入汉字了。

上述启动工作完成后，用 CTRL-L 键进行中英文方式转换，每按一次 CTRL-L 键，工作方式改变一次，进入中文操作后，在屏幕的左下方就会显示中文提示：区位、五形，用数字键表示汉字的区位码，用字母键表示五笔字型输入。

五笔字型汉字输入采用将汉字按顺序笔划拆成字根的方法，每个字根用一个字母键代表，一个汉字最多用四键输入。

汉字由字根组成，字根由笔划构成。笔划、字根、整字是汉字结构的三个层次。我们将一百多个字根按首笔笔画分为五类，如下表：

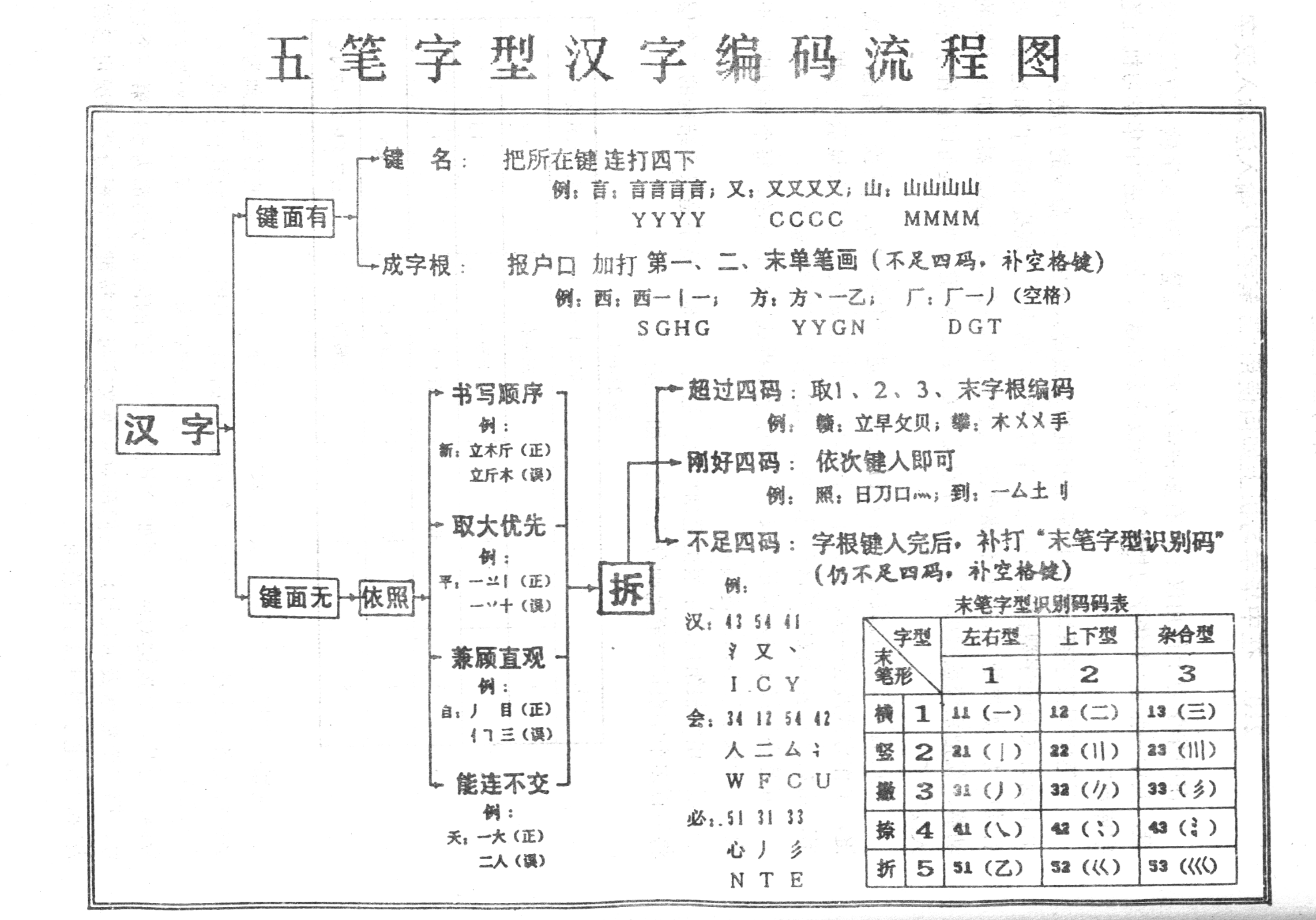
表 210-1

代号	笔划名称	笔划走向	笔划及其变形
1	横	左至右	一 ／
2	竖	上至下	丨亅
3	撇	右上至左下	丿
4	捺	左上至右下	＼丶
5	折	带转折	乙フ㇋㇄㇗㇉ㇷ㇆ㇾ

例如：字根“王”、“土”、“大”、“木”、“工”等分为一类，它们的首笔笔画为“横”。这五类字根每类各对应英文键盘上的一个区(五个按键)。再根据字根的其它特点将分类字根又分为五部分，每部分对应一个英文按键，每个按键称为一“位”。所以，字根都有相同的区位号码，与某一个按键对应。例如：字根“女”、“刀”的区位号码都为 53 (第一个数据表示区号，第二个数据表示位号)，它们都对应英文字母 N 按键。字根分类情况可参看“字根总表”和“五笔字型键盘字根总图”。

表 210-2 按助记符顺序排列的字根总表

五笔字型汉字编码流程图



末笔形 \ 字型		左右型 1	上下型 2	杂合型 3
横	1	11（一）	12（二）	13（三）
竖	2	21（丨）	22（‖）	23（川）
撇	3	31（丿）	32（〃）	33（彡）
捺	4	41（㇏）	42（冫）	43（氵）
折	5	51（乙）	52（巜）	53（巛）

五笔字型键盘字根总图

35 Q	34 W	33 E	32 R	31 T	41 Y	42 U	43 I	44 O	45 P
金钅鱼儿 勹犭乂 夕夕	人亻 八	月 用 彡 乃 豕	白 手扌 斤	禾 竹 丿 攵 彳	言讠文方 丶 广	立 六辛 冫 疒 门	水 氵 小	火 米 灬	之 辶 廴 冖 宀 礻

15 A	14 S	13 D	12 F	11 G	21 H	22 J	23 K	24 L	
工 匚 廾 廿 七 弋 戈	木 丁 西	大犬古石 三 厂	土士干 二 十 雨 寸	王 一 五 戋	目 丨 卜 上止	日 早 刂 虫	口 川	田 甲 囗 四 车 力	： ；

Z	55 X	54 C	53 V	52 B	51 N	25 M			
Z	纟 幺 勹 弓 匕	又 厶 巴 马	女 刀 九 巛 彐 臼	子 了 也 耳 阝 凵	已己巳 乙 尸 心 忄 羽	山 由 贝 冂 几	＜ ，	＞ 。	？ ／

五笔字型字根助记词

11 王旁青头戋（兼）五一，
12 土士二干十寸雨。
13 大犬三䍜(羊)古石厂，
14 木丁西，
15 工戈草头右框七。

21 目具上止卜虎皮，
22 日早两竖与虫依。
23 口与川，字根稀，
24 田甲方框四车力。
25 山由贝，下框几。

31 禾竹一撇双人立，
反文条头共三一。
32 白手看头三二斤，
33 月彡（衫）乃用家衣底。
34 人和八，三四里，
35 金勺缺点无尾鱼，
犬旁留乂儿一点夕，氏无七（妻）。

41 言文方广在四一，
高头一捺谁人去。
42 立辛两点六门疒，
43 水旁兴头小倒立。
44 火业头，四点米，
45 之宝盖，
摘礻（示）衤（衣）。

51 已半巳满不出己，
左框折尸心和羽。
52 子耳了也框向上。
53 女刀九臼山朝西。
54 又巴马，丢矢矣，
55 慈母无心弓和匕，
幼无力。

图 210—1 五笔字型键盘字根总图

3. 各类字根是按下述特点分为五个部分(五位):

(1) 位号与次笔号码一致，

例如：字根“王”：首笔“横”，号码为1，次笔“横”，号码为1，

所以在1区1位，区位号为“11”；

字根“之”：首笔“捺”，号码为4，次笔“折”，号码为5，

所以在4区5位，区位号为“45”；

字根“白”的区位号为32，

字根“文”的区位号为41，

字根“山”的区位号为25。

(2) 位号与笔划数目相一致，

例如： 字根“三”：笔划数目为三，区位号为13；

字根“水”：笔划数目为三，区位号为43；

字根“女”：笔划数目为三，区位号为53；

字根“灬”：笔划数目为四，区位号为44。

(3) 与主要字根形态相近或渊源一致时，其区位号与主要字根的区位号一样。例如：字根“冫く”、“冫丨く”、“氵”与主要字根“水”的形态相近，字根 “氵”与主要字根“水”渊源一致，因此这些字根的区位号同字根水的区位号一样，为43。

(4) 个别字根的区位号不具有上述特点，

例如：字根“车”、“力”、“心”等，这些字根需要摸索其它特点以帮助记忆，字根“心”最长的笔划为折笔，折笔内有一个点，故它的区位号为51；

字根″立″、″六″、″辛″、″疒″、″丬″、″丷″、″ツ″等字根首笔划为捺，而且均有两个点，故区位号为42。

4. 字根的记忆法

要使汉字输入的速度快，首先要记住每个字根的区位号码和按键的位置。要记住字根的区位号码，可在记住字根编号特点的基础上，将同一区的字根连起来，编成一首词，使之押韵上口有“诗”味。字根总图中的五首“诗”，每句第一个字对应键位的键名。

5. 汉字的拆分方法

汉字都是由一些字根组成。或者说，汉字中除了本身就是字根的以外，都可拆分或几个字根，汉字拆分字根的总原则是：按书写顺序，依次拆成为字根中最大字根，以增加一笔不能形成已有的最大字根来决定笔画的分组，直到把整个汉字拆分完毕。

拆分原则可归纳为以下四个要点：

(1) 取大优先

平： 一 丷 丨 (11 42 21)

重： 一日土 (31 11 22 12)

光：⺌ 儿 (43 35)

夷： 一弓人 (11 55 34)

勿： 勹丿丿 (35 32)

(2) 兼顾直观

舟：丿 丹 (31 33)

羊：丷 𰀁 (42 13)

(3) 能连不交

天：一大，不能拆作"二人"，因二者相交，(11 13)

于：一十，不能拆作"二丨"，因二者相交，(11 12)

丑:乙土，不能拆成"刀二"，因二者相交；(51 12)

(4) 能散不连

午:𠂉 十，都不是单笔画，应视上下关系，(31 12)

占:⺊ 口，都不是单笔画，应视上下关系，(21 23)

非：三 〢三，都不是单笔画，应视上下关系。(13 22 13)

汉字输入方法有以下几种：

1. 键名汉字输入方法

各键位左上角的黑体字根，叫该键的"键名"，如王，日，口等 25 个，它们的输入方法是将所对应的键连序按下四次，例如：

金:35 35 35 35 (QQQQ)

人:34 34 34 34 (WWWW)

大:13 13 13 13 (DDDD)

2. 成字字根输入方法

在字根总表中，除键名之外，本身就是汉字(包括"亻"、"氵"等国表码的部首在内)的字根叫成字字根，其输入方法是：先按该字根所在的键一下(叫报户口)，再按该字根的第一、第二及最末一个单笔画对应的键；例如：

方：方(41)报户口，(41)首笔，(11)次笔，(51)末笔；

用：用(13)报户口，(31)首笔，(51)次笔，(21)末笔；

石:石、一、丿、(13 11 31 11)　　氵:氵、丶、丶、(43 41 41 11)

3. 简码输入

常用的汉字中，多数只取其前边的一至三字根，再加空格输入即可。这就构成一，二，三级简码，一级简码只需按一个字根对应的键，这种简码也叫高频字码。例如:的:是需按 32(R)键和加按空格键即可。

二级简码需按二个与前两个字根对应的键。例如:李:按 14 52[S.B] 键和空格键。

三级简码需按三个与前三个字根对应的键。例如:想:按 14 21 51 [S.H.N]键和空格键。

有时一个汉字可有几种简码。例如：

经 55[X],55 54[XC],55 54 15[XCA]55 54 15 11[XCAG].

一级简码如下：

一:11[G],　地:12[F],　在:13[D],　要:14[S],　工:15[A],

上:21[H],　是:22[J],　中:23[K],　国:24[L],　同:25[M],

和:31[T]，　的:32[R]，　有:33[E]，　人:34[W]，　我:35[Q]，

主:41[Y]，　产:42[U]，　不:43[I]，　为:44[O]，　这:45[P]，

民:51[N]，　了:52[B]，　发:53[V]，　以:54[C]，　经:55[X]。

4. 超过四码字输入方法

组成汉字的字根个数超过四个，它的输入方法是：依次将它的一.二.三和第末个字根所对应的键按下即可。例如：

籍：　竹　三　小　日　(31 13 43 22)

魁：　白　儿　厶　十　(32 35 54 12)

5. 刚好四码字输入方法

组成汉字的字根个数刚好是四个时，它的输入方法是：依次将第一至四个字根所对应的键按下即可。例如：

涝：　氵.艹.冖.力　(43 15 45 24)

控：　扌.宀.八.工　(32 45 34 15)

6.__不足四码字输入方法

组成汉字的字根个数不够四个时，它的输入方法是先依次按下与字根对应的键，再补打“末笔字型识别码”，如果仍不足四个码，可补按空格键。

按构成汉字的各字根之间的位置关系，可以把汉字分为三种类型，如下表所示：(见表2)

表 210-3

字型代号	字型	图　示	字　例
1	左右	[illegible]	汉湘结封
2	上下	[illegible]	字莫花华
3	杂合	[illegible]	困凶这司乘本重天且

所消末笔字型识别码，就是由末笔划号与字型代号组合而成的号码。

例如：“汉”字的末笔字型识别码为 41，汉的末笔画捺的号码是 4，字型是左右型，号码为 1。

由此可见，确定不是四码字的输入号码。例如：

汉:43 54 41　　字:45 52 12　　华:34 55 12 22

拦:14 42 12 11　　磊:13 13 13 12　　抢:14 34 52 51

7. Z 键功能

Z 键称为万能学习键，它不但可以代替“识别码”，而且还可以代替您一时记不清或分解不准的任何字根，并通过提示行，使您不知道“力”在什么键上时，“劳”字可打成:“艹冖 Z　Z”，此时提示行显示：

1. 劳　AP　　2. 劳　APL　　3. 蓉　APW　　4. 荣　APS　　5. 莹　APG

这时，再按″1″或″2″键即可将″劳″字调到编辑位置。另外，提示还告诉您，“劳”字有一个二级简码 (AP)，即只打“艹冖”按空格键即可。还告诉您 (“劳 APL”)第三个键应是

“L”。

此外,还可用 Z 键查阅同一汉字有无简码,是几级简码 (按下第一个字根的对应键后再按三次 Z 键); 还可以查阅全部汉字的字典及其外码。简码情况 (按四次 Z 键)。

8. 重码与容错码

有相同编码的字叫″重码字″。例如:键入 ″YEU″, 即显示:

1. 衣　　　2. 哀

如需要″衣″字, 不必挑选, 只管输入下文, ″衣″字就会自动显示到编缉位置上, 如要显示″哀″字, 可打上一排数字键 ″2″。

对容易弄错编码的字, 容许按错码输入。例如:

长:丿　七　　　长:七　丿

长:丿　一　乙　　长:一乙　丿.

均可以把长字打出来。

在五笔型汉字系统下使用打印机 CTRL−P 键用于连接打印机开关,每按一次, 与打印机的连接状态就改变一次, 在程序中也可以用:

POKE　814 , 1 接通打印机

POKE　814 , 0 关闭打印机

CTRL−O 键用于显示或打印用户小字库的内容.显示时, 每个汉字前边还给出该汉字的区位码.用此命令, 用户可以随时查看机内小字库中的内容。

CTRL−Z 键用于打印字型转换开关, 在系统连接 9 针打印机时, 可打印出 16 种大小不同的字体, 用此命令可以随时在最大和最小字体之间切换, 在程序中可以用命令。

POKE 815,N　　　N＝0,1,2,——— ,15

CTRL−W 是造字功能键, 用此键可立即进入造字状态, 随后用可以随意设计, 编制自己定义的汉字, 图形或符号。

CTRL−B 键是清除机内小字库中的所有内容, 使用此键, 可对小字库进行清理, 更新, 也可在特殊要求下, 让程序与小字库分离后单独处理。

五笔字型汉字输入方法是一种科学的汉字编码方案, 它输入汉字的速度快, 方法易学, 在中华学习机上使用将会开发出更多的汉字软件。

八、CP／M 系统部分

211. CP / M 系统常用程序有哪些、如何在中华学习机上使用 CP / M 操作系统

CP／M 操作系统是为 CPU 为 Z80 或 8080 的计算机而设计的磁盘操作系统。由于中华学习机的 CPU 是 6502，所以不能够直接使用 CP／M 操作系统。

中华学习机是 APPLE Ⅱ机的兼容机，APPLE－Ⅱ机为能使用 CP／M 操作系统专门制做了 Z80 卡，当 APPLE Ⅱ机插上 Z80 卡后就能够运行 CP／M 操作系统了，那么中华学习机能不能直接使用 APPLE Ⅱ机的 Z80 卡呢？当然是可以的不过存在着以下问题：

1. Z80 卡太长不能直接插到中华学习机的扩展槽内，使用时要么打开上盖，要么用“过桥板”把扩展槽接到外边，使用不太方便。

2. 由于中华学习机只有一个扩展槽，当被 Z80 卡占用后，就无法再接打印卡了，这样就失去了输出打印功能。

为解决上述问题，目前已研制出中华学习机的专用卡 Z80／PRT 卡，该卡的电路吸取了原 Z80 卡的优点，并加以改进，使芯片有所简少，并使该卡增加了打印功能，从而有效的解决了上述两个问题。

当中华学习机插上 Z80／PRT 卡后，它的功能将进一步增强，大量优秀的 CP／M 系统下的软件都能在中华学习机上运行。特别直得一提的是 DBASE Ⅱ、Z80 宏汇编等在我国广为流行的软件都能在中华学习机上运行，从而使得中华学习不但是学习机，也可成为管理机和开发机。可见 CP／M 操作系统可使中华学习机展宽其应用领域，更好的为各行各业服务。

本书在介绍 CP／M 系统软件的地方，凡涉及到打印的问题，均是在 Z80／PRT 卡上实现的。

CP／M 操作系统，具有丰富的应用程序，这里只就常见的一些加以介绍：

1. APDOS.COM

功能：将 DOS3.3 格式的 T 类或 B 类文件转到 CP／M 格式的磁盘上。

2. ASM.COM

功能：8080 汇编程序。

命令格式：ASM [d:]fn.ext

3. ASMB.COM

功能：Z80 汇编程序。

命令格式：ASMB [d:]fn 程序扩展名必须是 Z80。

4. CAT.COM

功能：显示磁盘文件目录、文件所占用空间和磁盘剩余空间。

命令格式：CAT [d:]

5. CONFIGIO.BAS

功能：重新设置运行环境，如重新定义键盘，修改驱动模块等。

用法：见第 216 问。

6. COPY.COM

功能：磁盘拷贝。

命令格式：COPY <d1> = <d2>[／s]拷贝，若选任选项为只拷贝操作系统。

7. CPM56.COM

功能：将 44K 系统进行重定位，生成 56K 系统

8. CPM60.COM

功能:将 44K 系统或 56K 系统重定位生成 60K 系统。

9. DDT.COM

功能：对 8080 汇编程序进行动态调试。

命令格式：DDT[d:]fn.ext，ext 一般为 COM。当系统进入 DDT 环境，屏幕显示"–"表示：DDT 一切就绪。

DDT 包括下列子命令：

(1) A[addr]、直接汇编；脱离此命令为：″.″

(2) D[addr1，addr2]、显示内存单元命令。

(3) F addr1，addr2，n、填充命令；把 n 送到从 addr1 到 addr2 的单元中去。

(4) G addr1，addr2、执行从 addr1 开始到 addr2 之间的程序，并显示运行结果。

(5) I fn.ext、输入命令。将文件名送入文件控制块，准备由 R 命令读入内存。ext 为 COM 或 HEX。

(6) L[addr1[，addr2]]、反汇编命令。

(7) M addr1，addr2，addr3、块移动命令。将 addr1 至 addr2 之间的内容送到 addr3 开始的内存单元。

(8) R addr、读命令。将文件控制块所指的文件读入内存。起始地址为 addr+程序首地址。

(9) T[n]、跟踪命令。n 为跟踪的指令数。

(10) U[n]、同 T[n]，但不显示执行结果。

(11) S addr、向内存单元送数命令；退出该命令用″.″键。

(12) X[r]、显示所有寄存器或修改 r 寄存器的内容。

10. DEBUG.COM

功能：Z80 汇编程序动态检测。

命令格式：DEBUG[fn.ext]。

DEBUG 的系统提示符为"–"共有 19 个子命令。

(1) Aaddr 直接汇编命令。退出该命令用 ″.″ 键。

(2) B[addr1， addr2...]、设断点命令。addr1 等分别为断点地址，不选任选项为显示断点地址。

(3) BX [addr1， addr2...]、删除断点命令。

(4) C [n]、跟踪命令， n 为跟踪指令数，每执行一条指令显示其结果，执行中若遇到 CALL nn 命令时把它所调用的子程序当做一条命令看待。

(5) CN [n]、跟踪命令，做用同(4)，但不显示执行结果。

(6) CJ、转移跟踪命令。遇有 JP、JR、CALL 等转移指令时中断程序运行。

(7) D [{ <addr> / <addr1, addr2> / <addr1, Sn> }]、存贮器显示命令。

若选择 addr，则从 addr 地址开始显示。若选择 addr1， addr2 则显示从 addr1 到 addr2 之间的内容。若选择 addr1， Sn 则显示从 addr1 开始到 addr1+Sn 结束的内容。

(8) DR、显示寄存器命令。R 指 A、B、C、D、E、H、L、CP、SP 等寄存器。

(9) G [addr1， addr2]、执行命令。程序的启动首地址为 addr1，结束地址为 addr2。

(10) H<表达式>、计算命令。可计算表达式的值并显示。

(11) L [{ <addr> / <addr1, addr2> / <addr, Sn> / < , addr> / <Sn> }]、反汇编命令。

任选项见(7)。

(12) M <addr1， addr2， addr3>、块移动命令。把从 addr1 到 addr2 之间的内容移到 addr3 开始的地址中去。

(13) Q { <addr1，addr2，m> / <addr，Sn，m> }、检索命令。在 addr1 到 addr2 或从 addr 到 addr+Sn 之间检索 m，并显示 m 所在单元地址。

(14) S { <r> / <addr> }、置数命令。可向寄存器或地址置数。脱离命令为"."

(15) T [n]、跟踪命令。跟踪程序运行并显示执行结果， n 为指令数。

(16) TN [n]、跟踪命令。功能同 T[n]，但不显示执行结果。

(17) TJ、转移跟踪命令。遇到转移指令 JP、JR、CALL 时，中断执行并显示执行结果。

(18) V { <addr1， addr2， addr3> / <addr， Sn， addr3> }、数据块比较命令。

将从 addr1 到 addr2、或从 addr 到 addr+Sn 之间的数与 addr3 开始的数据进行比较，并显示出内容不同的单元地址。

(19) Z { <addr1，addr2，m> / <addr， Sn， m> }、填充命令。向指定范围的内存单元填充数据 m。

11. DOWNLOAD.COM

功能：可将不同机种 CP／M 系统的文件通过 RS232 串行通讯接口卡载入苹果机 CP／M 系统，并存入磁盘。苹果机 RS232 接口卡应插在 2 号槽。(中华学习机不能直接使用)

12. DUMP.COM

功能：以十六进制代码的形式显示文件内容。

命令格式：DUMP <fn.ext>

13. ED.COM

功能：行编辑程序，可用来编辑、输入文件和程序。

命令格式：见第 237 问。

14. EX.COM

功能：用以协调某些 CP／M 系统程序在中华学习机上运行时，显示不兼容的问题，如 ASMB.COM 在运行后不能显示汇编出错的总数。若用 EX 文件加以协调就可解决。

命令格式：EX <系统文件名>

15. FORMAT.COM

功能：磁盘格式化程序。

命令格式：FORMAT [d:] 该命令能破坏磁盘数据，使用时应当注意。

16. L80.COM

功能：连接程序。可使由 ASMB、FORTRAN-80、COLOB 等语言编译后产生的目标代码文件进行连接定位，生成可运行的文件。

命令格式：L80 <代码文件名>[／开关]

其中开关包括：

(1) R 复位，使程序回到初始状态。

(2) E 返回操作系统。

(3) G 执行连接后的程序。

17. LOAD.COM

功能：将可执行文件装入存贮器。

命令格式：LOAD <fn.ext>扩展名应为 COM。

18. MFT.COM

功能：在单驱动器配置下拷贝文件。

命令格式：见第 235 问。

19. PIP.COM

功能：可完成文件的拷贝、打印等工作。

命令格式：见第 238 问。

20. RW13.COM

功能：可读写 13 扇区格式的磁盘。

命令格式：RW13

21. STAT.COM

功能：显示、修改系统状态。

命令格式：见第 239 问。

22. SUBMIT.COM

功能：执行批处理命令文件。

命令格式：见第 217 问。

23. XSUB.COM

功能：使批处理文件能够带数据参数。

用法：该命令应放在批命令文件的第一行具体方法见第 217 问。

此外 CP／M 系统还有许多高级语言文件如：BASIC-80， FORTRAN-80，COBOL-80 以及 DABSE-Ⅱ等，在这里就不一一例举了。

212. 怎样使 CP／M 系统能够利用中华学习机的全部 RAM 资源

MICROSOFT 公司为苹果机配置的 CP／M 操作系统一般只支持 44K 的内存资源，而中华学习机的内存为 64K，如果仍用原 44K 的 CP／M 操作系统对于内存资源无疑是一种浪费，为有效的利用内存，可在 CP／M 系统被引导后运行 CP／M56 或 CP／M60 程序。方法如下：

A > CPM56　　或　　A > CPM60

然后关机，再开机重新引导，系统会显示出：

APPLE Ⅱ　CP／M	或：APPLE Ⅱ　CP／M
56K　VER 2.20	60K　VER　2.20
(C)　1980　MICROSOFT	(C)　1980　MICROSOFT

通过运行 CPM56 或 CPM60 命令，软件修改了操作系统有关信息，使系统对内存的管理能力有所提高。修改后，CP／M 系统各模块地址变化情况如下：

模块名称	44K 系统	56K 系统
CCP	9400H	C400H
BDOS	9C00H	CC00H
BIOS	AA00H	DA00H
RAM 底部	AFFFH	DFFFH

213. CP／M 操作系统内存地址是怎样分配的

CP／M 操作系统在工作时需要连续的地址空间，但中华学习机(包括苹果机)工作时内存地址不能连续使用，如：监控程序要占用地址空间第 0 页($0000－$00FF)，系统堆栈要占用第一页($0100—01FF)，键盘缓冲区要占用第二页($0200－$02FF)，文本显示区要占用 $0400－$0700、外设接口要占用 $C000－$CFFF 等，把内存空间分成了好几块。

为解决 CP／M 系统地址连续问题，在 CEC－Z80 卡上设计了一个地址转换电路，把 6502 CPU 的地址空间加以转换，使从 Z80 CPU 角度上看，地址空间达到连续。地址的转换关系如下：

Z80　地址	6502　地址
0000H—OFFFH	$1000－$1FFF
1000H—1FFF	$2000－$2FFF
2000H—2FFFH	$3000－$3FFF
3000H—3FFFH	$4000－$4FFF
4000H—4FFFH	$5000－$5FFF
5000H—5FFFH	$6000－$6FFF
6000H—6FFFH	$7000－$7FFF
7000H—7FFFH	$8000－$8FFF

8000H–8FFFH　　$9000–$9FFF
9000H–9FFFH　　$A000–$AFFF
A000H–AFFFH　　$B000–$BFFF
B000H–BFFFH　　$D000–$DFFF
C000H–CFFFH　　$E000–$EFFF
D000H–DFFFH　　$F000–$FFFF
E000H–EFFFH　　$C000–$CFFF
F000H–FFFFH　　$0000–$0FFF

了解上述转换关系是非常有用的，根据对应关系，了解 6502 地址的用户就可以在 CP／M 系统下调用 6502 的监控程序，完成两个系统间的数据传输，使系统间互相取长补短，充分发挥应有的功能。

214. 如何从 CP／M 系统直接启动 DOS 盘

在使用完 CP／M 操作系统后，再接着使用 DOS3.3，最常用的方法是重新开机，把 DOS 系统引入。此方法固然方便，但也有不足。如当需要完成 CP／M 系统与 DOS3.3 系统间的数据传输时就不方便了，因为一关机数据就会丢失。

为解决在不关机的情况下由 CP／M 系统直接启动 DOS 盘问题，可利用下列程序帮助解决。

程序清单：

```
LD   (HL), A
LD   C, 1
CALL  5
LD   HL, C600H
LD   (0F3D0H), HL
LD   HL, (0F3DEH)
```

输入方法：

键入：DEBUG、A、输入程序、.、CTRL–C、SAVE 2 CPMTODOS.COM，至此程序输入结束，然后键入 CPMTODOS 程序起动等待输入一个字符，当按任意键后，A 盘转动，此时 DOS3.3 盘若已装入驱动器机就能够被引导。

215. CP／M 操作系统有哪些系统调用，应如何使用

操作系统不但能够有效的利用计算机资源，合理的管理计算机资源，还为用户使用计算机提供了方便的条件，这集中表现在两个方面，其一，提供了一些简单的命令，用户只要按几个键，就能够完成如：显示、存盘、打印等工作。其二，为用户提供了最基本、最常用的子程序如：输入一个字符、显示一个字符、打开一个文件等，使用户的编程工作大为减化。

所谓系统调用就是调用操作系统提供的子程序。操作系统为方便调用把所有子程序编

号，规定了入口参数，并且所有调用从同一个入口进入，系统会根据入口参数来决定转向那个子程序，这样使得系统调用非常简单而易于掌握。

系统调用命令格式: CALL 5.

入口参数: 一般有 1 或 2 个。一个是调用号，为必有的，放在 C 寄存器中; 另一个参数视需要而定。若有则放在寄存器对 DE 或 BC 中。

如:

```
LD      C， 09H
LD      DE， 0800H
```

出口参数: 出口参数为子程序运行后带回的结果。

0 号调用

功能: 系统复位

入参: C＝00H; 入参即入口参数，以下相同。

出参无; 出参即出口参数，以下相同。

1 号调用

功能: 键盘输入并显示

入参: C＝01H

出参: (A)＝输入字符的 ASCII 码。

2 号调用

功能: 显示一个字符

入参: E＝输出字符的 ASCII 码; C＝02H。

出参: 无。

3 号调用

功能: 从纸带机读入一个字符。

入参: C＝03H

出参: A＝输入字符的 ASCII 码。

4 号调用

功能: 向穿孔机输出一个字符。

入参: C＝04H, E＝输出字符的 ASCII 码

出参: 无

5 号调用

功能: 向打印机输出一个字符。

入参: C＝05H、 E＝输出字符的 ASCII 码。

出参; 无

6 号调用

功能: 控制台输入输出

入参: C＝06H，当 E＝0FFH 时为输入; E＝其它字符时为输出字符的 ASCII 码。

出参: 输入时: A＝00 表示输入设备未准备就绪， A≠00 时为输入字符的 ASCII 码。

输出时: 无。

7 号调用

功能：读取 I／O 字节。

入参：C＝07H。

出参：A＝I／O 字节的值。

8 号调用

功能：置 I／O 字节。

入参：C＝08H、E＝I／O 字节值。

出参：无。

9 号调用

功能：打印字符串。

入参：C＝09H，DE＝字符串首址。字符串最后一个字符应为 $ 。

出参：无。

10 号调用

功能：输入字符到缓冲区并显示。

入参：C＝0AH、DE＝缓冲区首地址。

出参：无。

11 号调用

功能：取控制台状态。

入参：C＝0BH

出参：A＝状态，当状态＝FF 表示已输入，状态＝00 表示未输入。

12 号调用

功能：取系统版本号。

入参：C＝0CH

出参：HL＝版本号。其中 H＝00 为 CP／M、01 为 MP／M，L＝00 为 2.0 以前的版本，20、21…表示 2.0 以后的版本。

13 号调用

功能：磁盘系统复位。

入参：C＝0DH

出参：无。

14 号调用

功能：选择磁盘。

入参：C＝0EH、H＝所选磁盘号。00 为 A 盘、01 为 B 盘。

15 号调用

功能：打开文件。

入参：C＝0FH，DE＝文件控制块 FCB 的首块。

出参：A＝目录项编号，当 A＝FF 表示文件未找到。

16 号调用

功能：关闭文件。

入参：C＝10H，DE＝FCB 首地址。

出参：A＝目录项编号，当 A＝FF 表示关闭文件失败。

17 号调用

功能: 在第一项目录中查找文件。

入参: C＝11H、DE＝FCB 首地址。

出参: A＝目录项编号，若 A＝FF 表示文件设找到。

18 号调用

功能: 在下一项目录中查找文件。

入参: C＝12H。

出参: A＝目录项编号，若 A＝FF 表示文件未找到。

19 号调用

功能: 删除文件。

入参: C＝13H， DE＝FCB 首址。

出参: A＝目录项编号，若 A＝FF 表示文件未找到。

20 号调用

功能: 顺序读下一个记录。

入参: C＝14H， DE＝FCB 首址。

出参: A＝目录项编号，若 A＝00 表示读成功。

21 号调用

功能: 随机写入一个记录。

入参: C＝15H、DE＝FCB 首址。

出参: A＝00H 表示写入成功； A＝01H 表示目录项满； A＝02H 表示磁盘满。

22 号调用

功能: 建立文件。

入参: C＝16H、DE＝FCB 首址。

出参: A＝目录码， A＝FF 表示磁盘已满。

23 号调用

功能: 更改文件名。

入参: C＝17H、DE＝FCB 首址。

出参: A＝00–03H，表示更改成功；A＝0FFH 表示文件没找到。

24 号调用

功能: 取磁盘登录字节。

入参: C＝18H

出参: 在 HL 中，从 L 的最低位到 H 的最高位依次表示: A–P 共 16 个槽号，该位为 0 表示未登录，为 1 表示已登录， A＝L。

25 号调用

功能: 取当前盘号。

入参: C＝19H

出参: A＝当前盘的盘号。

26 号调用

功能: 设置缓冲区。

入参: C＝1AH， DE＝缓冲区首址。

出参: 无。

27 号调用

功能: 取当前盘图。

入参: C＝1BH。

出参: BC＝盘图区首址。

28 号调用

功能: 置当前磁盘写保护。

入参: C＝1CH

出参: 无。

29 号调用

功能: 取只读字节。

入参: C＝1DH。

出参: HL＝只读字节，共 16 位分别表示 16 个槽号，某位为 1 表示只读。

30 号调用

功能: 设置文件属性。

入参: C＝1EH， DE＝FCB 地址。

出参: A＝目录码。

31 号调用

功能: 取磁盘参数表地址。

入参: C＝1FH

出参: HL＝磁盘参数表地址。

32 号调用

功能: 设置、取回用户号。

入参: C＝20H、0FFH 表示取， E＝用户号表示置。

出参: A＝当前用户号。

33 号调用

功能: 随机读取一条记录。

入参: C＝21H、DE＝FCB 首址。

出参: A＝返回码， 00 表示读取成功。

34 号调用

功能: 随机写一条记录。

入参: C＝22H、DE＝FCB 首址。

出参: A＝返回码。

35 号调用

功能: 计算文件大小。

入参: C＝23H， DE＝FCB 首址。

出参: 将文件大小写入 FCB 的第 33–35 位。

36 号调用

功能：设定随机记录。

入参：C＝24H，DE＝FCB 首址。

出参：使记录定位。

例：输入字符串到 2000H 单元。

```
LD DE， 2000H
LD C， 0AH
CALL 5
JP 0000H
```

216. 在 CP／M 系统中如何显示小写字母

中华学习机本身有输入和显示小写字母的功能，但在 CP／M 系统中并不给与承认，所以不用够直接输入小写字母。

如需输入或显示小写字母时需通过 CONFIGIO 程序对系统配置进行修改。其方法如下：

开机将 CP／M 系统引入，然后键入 MBASIC CONFIGIO 这时屏幕上会显示出：

CAN YOUR APPLE DISPLAY LOWER CASE(Y／N)?

接着键入"Y"，屏幕上就显示出包括小写字母在内的 CONFIGIO 程序的功能菜单，说明小写字母显示功能已装入，再按"Q"键，退出 CONFIGIO 程序。以后只要按下 CAP-LOCK 键，就可以输入和显示小写字母了。

217. 怎样利用批处理程序使调试程序过程自动化

在 CP／M 操作系统环境下调试程序每次往往需要重复使用多种命令，如调试 Z80 汇编就要用 ED、ASMB、L80 等命令。为减化操作，可利用系统的批命令处理功能，使调试过程自动化。批命令处理的格式为：

SUBMIT <fn>[Parm1 Parm2...Parmn]

其中 fn 为当前盘的批处理文件名，文件扩展名应为：".SUB"，如：CF.SUB，批处理命令文件的内容由系统命令序列组成，按使用的先后顺序排列。Parm 1、Parm 2...Parm n 是命令参数，它们与文件中的形式参数相对应，形参的符号是：＄1、＄2...＄n，批处理命令执行期间系统会自动把参数的内容传递给形式参数，传递规则为：Parm 1 送＄1、Parm 2 送＄2，等，若文件中没有形参，命令参数可以省略。批命令文件的建立可依靠 ED 或 PIP<fn.ext>＝CON：命令来完全。

例：建立一个调试 Z80 汇编程序的批处理命令文件。文件名为：AMS.SUB。内容如下：

```
ED    $1.Z80
ERA   $1.BAK
ASMB  $1
PIP   PRN:=$1.PTN
```

调用该文件可键入：

SUBMIT　ASM POI　　其结果相当于执行了

ED　POI.Z80

ERA　POI.BAK

ASMB POI

PIP　PRN:＝POI.PRN

命令。如果想再调试另一个程序，只要更换 SUMBIT 命令中的参数 POI 既可。

在使用 SUBMIT 命令时，当形式参数过多时，命令参数就必然曾加，使命令行变的很长，在输入时容易出错。为解决这个问题，系统还提供了 XSUB 命令，该命令可以把 SUBMIT 命令中不变的参数做为常数直接写到批处理文件中，使命令参数的数量减少。XSUB 命令应出现在批处理文件的第一行。

例：XSUB

ED　$ 1.Z80

#A

#T

在执行时#A 和#T 会分别送入 ED 程序。相当于键入#A、#T，从而实现了运行过程自动化。

218.　有些 CP／M 系统为何不能启动打印机、应如何解决

系统不能启动打印机的原因很多，既有硬件的原因也有软件的问题。较为常见的一种是由于操作系统本身原因引起的。

当 44K 系统升级为 56K 系统时需使用 CPM56.COM 软件，该软件在对原系统进行修改时，逻辑设备到物理设备转换时有误，使 CTRL−P 后不能正常启动打印机。

修改方法：

在 44K 系统下键入：DDT、CPM56、S272F、31，其目的是把地址 272F 单元原来的 3E 改为 31，修改后键入 CTRL−C、SAVE 42 CPM56.COM 把修改后的内容存盘。这时再用修改后的 CP／M56 升级 44K 系统就可以正确连接打印机了。

219.　如何把几个文件连成一个

在程序设计过程中，经常需要把几个子程序顺序连接起来，形成一个完整的程序。这一连接过程可以由 PIP 命令来完成。连接的命令格式为：

PIP DA:fn1.ext＝d:fn2.ext，　d:fn3.ext，　...d:fnn.ext 其功能为：把等号右面的文件 fn2.ext、fn3.ext...按顺序连接起来，存入 fn1.ext 中，　d：为任意驱动器。在连接 BASIC−80 程序时，应注意各程序的行号不要相重，避免连接后修改过多。另外：连接的程序必须是文本文件。

例：连接 BASIC 程序 A1.BAS 和 A2.BAS 到 A.BAS 中。

PIP A.BAS＝A1.BAS，A2.BAS

220. 如何把 Z80 的指令代码通过 EPROM 写入卡写入 EPROM

Z80 汇编是系统开发中常用的语言，中华学习机在 CP／M 系统下，从编辑、汇编、连接到动态调试为开发提供了一系列手段和良好的工作环境。但是，由于在 CP／M 系统方式下，不能直接利用 EPROM 写卡把编好的程序写入 EPROM，所以在开发的范围上受到了一定的限制。这是因为 EPROM 写入卡是为 6502cpu 开发的，只能在 APPLESOFT 方式下使用。

下面介绍一种简单的方法来解决上述问题。

(1) 开机后对程序进行输入、修改、汇编、连接行成 Z80 指令代码然后存盘。

(2) 用 DDT 程序，把 Z80 指令代码文件装入内存，命令为：

DDT 文件名.COM 当看到"–"提示符后依次输入：

```
–SF3D2   ××  00
F3D3     ××  00
F3E4     ××  00
F3D5     ××  .
–SF3D2   00  4C
       00  69
       00  FF
       00  .
   *
```

"＊"号为监控提示符，说时系统已进入监控程序。

(3) 取出 CP／M 盘、插入 DOS3.3 盘键入：6、CTRL–P 引入 DOS、用 BSAVE 命令把 Z80 代码存盘，此时应注意 Z80 地址与 6502 地址的映射关系。

(4) 关机后拔下 CEC–Z80 卡，插上 EPROM 写入、卡拔好写入卡的开关。

(5) 重新开机，用 BLOAD 命令把 Z80 命令代码文件装入内存相应位置。

(6) 按入 PR#1 启动 EPROM 写入卡，然后按菜单提示即可完成写入工作。

221. CP / M 的 BASIC–80 与 APPLESOFT 有哪些不同

为了使了解中华学习机 APPLESOFT–BASIC 语言的读者更快的了解 BASIC–80 语言的特点，在这里为读者列出 BASIC–80 与 APPLESOFT–BASIC 不同点，以供参考：

APPLESOFT BASIC 没有的命令：

AUTO 自动编行号命令。语法格式为：

AUTO [<行号>，[<增量>]]。

BEEP 产生音调和拍节。

格式：BEEP<音调>；<拍节>；0≤音调≤255；0≤拍节≤255

COMMON 把指定变量存入公共区。

格式：COMMON <变量名表>

DEF　SNG ／ DBL ／ STR　定义变量为单精度、双精度和字符串。

格式: DEFD　SNG 单精度; DEF　DBL　双精度; DEF　STR　字符串

DEF　USR　定义调用汇编语言首地址。

格式: DEF　USR　<数字>=<整数表达式>

EDIT　行编辑命令。

格式: EDIT <行号>

ERASE　取消已定义的数组。

格式: ERASE　<数组名表>

ERR、ERL 错误码和错误命令行号。

用法: 用于　IF　THEN 语句，控制程序流向。

ERROR: 模拟出错命令。

格式: ERROR <整数表达式>

FIELD: 定义文件字段宽度。

格式: FIELD #<文件通道号>，<字段宽度 1> AS<字段名 1>，<字段宽度 2>AS<字段名 2>…。

FILES:　显示文件目录。

格式: FILES。

KILL:　删除指定的文件。

格式: KILL <FN.EXT>

LLIST:　打印程序清单。

格式:　LLIST　[<行号>[.[<行号>]]]

LPRINT:　打印语句。

格式: LPRINT [<表达式 >]

LPRINT／PRINT　USING: 按特定格式输出。

格式:　LPRINT　USING <字符串描述符>; <表达式>

　　　PRINT

其中字符串描述符有下列定义:

"!": 只输出一个字符。

"\n 空格\": 输出 n+2 个字符。

"&": 输出长度与输入长度相等。

"#": 指定输出字符的位置。

"+": 置于字符串前面或后面，会在输出数据的前面或后面显示数值的符号。

"–": 同"+"，但数值为正时不显示符号。

"*": 代表数字位置，当数字位置比 * 号个数少时，用 * 号填充

"$": 代表数字，并在数字左边输出"$"

↑↑↑↑: 以指数格式输出数字。

"–": 在字符串中使用，使下一个字符紧接着上一个字符。

"%": 当数字长宽大于给定长度，则在数字左边印出%。

LSET／RSET:　左／右对齐。

格式：{LSET
RSET}<字符串变量>=<字符串表达式>

MERGE 把一指定的磁盘文件与内存中的程序合并。

格式：MERGE<文件名称>

RANDOMIZE：启动随机数。

格式：RANDOMIZE

RESET：换盘后重新调入磁盘分配图。

格式：RESET。

SWAP：数据交换。

格式：SWAP <变量 1>，<变量 2>.

SYSTEM：退出 BASIC-80 返回 CP/M 系统.

格式：SYSTEM.

WAIT：检测地址情况.

格式：WAIT <地址>，I [，J]；若地址内容 XOR J AND I=0 则对该地址继续检测；若不为零，则顺序执行.

WHILE...WEND 当型循环.

格式：WHILE <条件>

<循环体>

WEND

WIDTH：定义设备输出宽度、或为屏幕开窗口.

格式：WIDTH [LPRINT] <行宽>.

WIDTH [<行宽>]，[<屏幕高度>].

还有一些命令在功能上有所加强，例如:IF...THEN...ELSE 增加了在否定情况下的处理。有些语句虽随形式相同但功能完全不同，如 GET 语句等.

了解了上述特点，读者一定不难写出具有 BASIC-80 风格的程序来。

222. MBASIC 与 GBASIC 有何区别

MBASIC 与 GBASIC 统称 BASIC-80 它们的语句基本相同，只是在作图语句方面有些差异。MBASIC 支持低分辨率作图，而 GBASIC 支持高分辨率作图。

在作图时，MBASIC 有二种屏幕显示方式，一种为:40＊40 个点另加 4 行文本区；另一种为 40＊48 个点。MBASIC 的作图命令有：

COLOR=<颜色代码>

功能:置屏幕颜色，颜色代码与颜色的对应关系：

0 黑	4 暗绿	8 棕	12 绿
1 紫红	5 灰	9 橙	13 黄
2 暗蓝	6 中蓝	10 灰	14 水色(agua)
3 紫	7 淡蓝	11 粉红	15 白

GR [显示方式代码]，[颜色代码]

功能:设置屏幕显示方式。

显示代码为 0 或 1，其含意为:

代码	显示方式
0	40 * 40 点加 4 行文本区。
1	40 * 48 点。

如省略显示方式代码则为 0。

颜色代码与 COLOR 语句相同，如省略则为黑色。

HLIN <X1>，<X2> AT <Y>；0⩽X1、X2⩽39，0⩽Y⩽47

功能:在作图方式下，<X1，Y>点与<X2，Y>点之间划一条线。

GBASIC 在作图方式下也有两种屏幕方式，一种为:280 * 160 点+4 行文本行，另一种为:280 * 192 点。GBASIC 的作图语句有:

HCOLOR = <颜色代码>，

功能:设置高分辨率做图时的屏幕颜色

代码对应的颜色为:

0 黑	3 白	6 蓝	9 白 1	12 补色
1 绿	4 黑	7 白	10 黑 2	
2 紫	5 橙	8 黑 1	11 白 2	

HGR <显示方式代码>，<颜色代码>

功能:设置高分辨率图时屏幕显示方式。

显示方式代码与显示方式关系如下:

代码	显示方式	是否清屏
0	280 * 160 点+4 行文本	是
1	280 * 192 点	是
2	280 * 160 点+4 行文本	否
3	280 * 192 点	否

颜色代码同 HCOLOR 语句

HPLOT [<X1>，<Y1>] [TO <X2>，<Y2>…[TO <Xn>，<Yn>]]

其中:0⩽Xn⩽279;0⩽Yn⩽191

功能:在高分辨图形方式下，划出点(X1，Y1)、(X2，Y2)、…(Xn，Yn)之间的连线。

HSCRN(X，Y)

功能:当(X，Y)坐标处划有点时，语句值为 1，否则为 0。

MBASIC 与 GBASIC 除在上述语句有所区别外，用户的程序空间也不相同，在 44K CP／M 系统中，MBASIC 的用户空间为:13.8K，GBASIC 的用户空间为 5K。在 56K CP／M 中 MBASIC 为:25.8K，GBASIC 为:17K。

223. 在 BASIC–80 中如何使用编辑命令

在 BASIC–80 中配置了功能较强的行编辑命令 EDIT，通过该命令，可以对 BASIC

程序进行编辑修改，为调试程序提供了方便。

命令格式：EDIT<行号>

在编辑状态下，有 5 类子命令，其功能如下：

1.移动光标命令：

[n]SPACE 键：光标向右移动 n 列，并显示出原来的内容，当 n 省略时值为 1。

←键：光标向左移动。

2.插入新内容：

I：在光标处开始插入。

X：在行末处插入。

ESC:结束插入方式。

RETURN：结束插入方式，并退出编辑状态。

3.删除指定内容：

[n]D:删除从光标右侧开始 n 个字符，并把删除内容两端用"＼"符号括起来，当 n 省略时为 1。

H：删除光标右侧字符并进入插入方式。

4.检索。

[n]S<目标字符>：检索第 n 次出现的目标字符，并显示该字符前面所有内容，如未找到，光标移至行末。

[n]K<目标字符>：同 S 子命令，当查到后将目标字符删除。

5.更换内容。

[n]C<更换字符>：从下一个字符开始更换 n 个字符。

另外还有一些控制子命令。

RETURN：修改完毕显示该行全部内容并退出。

E:修改完毕并退出。

Q:放弃修改并退出。

L:修改完毕光标移至行首。

A:放弃修改但不退出。

当 BASIC-80 系统解释 BASIC 程序时，一但发现句法错误自动进入 EDIT 编辑状态等待用户修改。

224. 如何建立磁盘文件

BASIC-80 的磁盘文件管理系统比 DOS3.3 的文件管理有许多优点。它建立文件方便，使用灵活规范，特别是与 IBM／PC 机的 BASIC 很相似，是很值得一用的。

BASIC-80 的数据文件有两种类型，顺序文件和随机文件。所谓顺序文件是在读写时只能按记录顺序从前向后逐条进行，不能跳跃。而随机文件文件则不受此限制。下面分别给与介绍。

(1) 顺序文件

有关顺序文件的语句如下：

OPEN <“方式”>，#<通道号>， <“文件名”>

功能：按指定方式打开文件，并规定通道号。其中：“方式”为“I”表示：输入，这时，只能把数据从磁盘读入内存。“方式”为“O”表示输出，数据由内存写入磁盘。“通道号”为：正整数。“文件名”表示被打开的文件。

CLOSE #<通道号>。

功能：关闭指定通道号所代表的文件。

INPUT #<通道号> <变量表>。

功能：从通道号所指定的文件中输入一个数据并赋给变量表中的变量。

LINE INPUT #<通道号> <字符串变量>。

功能：从通道号所指的文件中输入一行数据并赋给字符串变量。

WRITE #<通道号>， <表达式>。

功能：把表达式的内容写入通道号所指的文件，并且当表达式为数值时，数据之间加“，”号；当表达式为字符串时，数据两端加双引号。

PRINT #<通道号> <表达式>

功能：同 WRITE 语句，只是未加逗号或引号。

EOF 〈通道号〉

功能：测试该道所指文件是否已经读完，当读完时函值为“真”，否则为“假”。

LOC

功能：给出被打开文件所占用的扇区数，一个扇区为 128 个字节。

例 1：建立一个顺序文件。

```
10 OPEN "O", #1, "DATA"
15 INPUT "NAME:", N$
20 IF N$ =CHR$(13) THEN 45
25 INPUT "AGE:", A$
30 INPUT "SEX:", S$
35 PRINT #1, N$, A$, S$
40 GOTO 15
45 CLOSE #1
50 END
```

例 2:读取上例文件中的数据：

```
10 OPEN "I", #1, "DATA"
15 IF EOF(1) THEN 35
20 INPUT #1, N$, A$, S$
25 PRINT N$, A$, S$
30 GOTO 15
35 CLOSD #1
40 END
```

OPEN "R" #<通道号>，“文件名”， <记录长度> 其中：“R”指随机方式；记录长度指一条记录所占用的字节数。

功能：建立或打开指定的随机文件。

FIELD ＃<通道号> <宽度> AS <字段 1>，<宽度 2> AS <字段>…。

功能：定义记录中字段的数量以及每个字段的宽度。各字段度的总和应等于记录长度。

LSET / RSET <字段名> = 字符串变量。

功能：把字符串变量的内容送入字段时，若字符串变量的长度小于字段长度，LSET 表示从左端送入，右端填充空格；REST 表示从右端送入左端填充空格。

GET ＃<通道号>，<记录号>

功能：将通道号所指的随机文件的，记录号所对应的数据读入缓冲区。

PUT ＃<通道号>，<记录号>

功能：把缓冲区中的记录写入通道号所指文件的相应记录中。

LOC(<通道号>)

功能：给出通道号所指记录的当前记录号。

MKI $、MKS $、MKD $

功能：数据在写入缓冲区以前必须转化为字符串类型。MKI $：可把整型转化为字符串；MKS $：把单精度型转化为字符串；MKD $：可把双精度型转化为字符串。

CVI、CVS、CVD

功能：把缓冲区内的字符型数据转化成数值型数据。CVI：把字符型转化成整型；CVS：把字符型转化成单精度型；CVD：把字符型转化成双精度型。

例 3.建立随机文件。

```
10 OPEN "R", #1, "FILE", 32
20 FIELD #1, 20 AS N $, 4 AS A $, 8 AS P $
30 INPUT "ENTER CODE ";CODE%
35 INPUT "NAME" X $
40 IF X $ =CHR $ (12) THEN 120
45 INPUT "NAME" X $
50 INPUT "AMOUNT";AMT
60 INPUT "PHONE";TEL $
70 LSET N $ =X $
80 LSET P $ =TEL $
90 LSET A $ =MKS $ (AMT)
100 PUT #1, CODE%
110 GOTO 30
120 CLOSE #1
130 END
```

例 4：输出 FLIE 文件的内容。

```
10 OPEN "R", #1, "FILE", 32
20 FIELD #1, 20 AS N $, 4 AS A $, 8 AS P $
30 INPUT "ENTER CODE ";CODE%
```

```
35 IF CODE% = 0 THEN 70
40 GET #1, CODE%
50 PRINT N$, CVS(A$), P$
60 PRINT :GOTO 30
70 CLOSE #1
80 END
```

225. 如何重编 BASIC-80 程序的行号

对程序重编行号，是编程过程中，经常进行的工作，BASIC-80 为用户提供了这一功能。

命令格式：RENUM [首行号，] [需改程序首行号，] [改后行号增量]

功能:对当前程序重编行号，若省略选择项系统默认：从当前程序的第一行开始，重编首行号为 10；行号增量为 10。否则系统从[需改程序首行号]开始重编，如果只输入两个数据，系统自动认为是新的首行号和行号增量，并对全部程序进行修改。

RENUM 命令执行后程序的最大行号不能大于 65529，否则程序出错。

226. 在 BASIC-80 中一个运行程序如何调用另一个程序

由于 BASIC-80 提供给用户的内存容量有限，在运行较大的程序时，内存常常不够用。解决这一问题的方法是把程序分成若干段，存在磁盘上，每段都是相对独立的程序。在运行时，用到哪段，就把哪段调入内存，程序调入时，覆盖以前的内容，这样，就可以在内存容量不变的情况下运行较大的程序了。

要完成上述功能，必须先解决两个问题，第一，怎样连接各段程序？第二，怎样在程序间实现变量传递。BASIC-80 为此专设了两个语句以解决上述问题。语句格式分别为：

CHAIN [MERGE]fn [, [行号], ALL] [DELETE 范围]

功能：将文件名为 fn 的 BASIC 程序调入内存，当选择任选项[MERGE]时，调入的程序与内存中的程序拼接，且不破坏内存中的程序。否则，覆盖内存程序；fn 为 BASIC 程序名；“行号”表示调入程序的入口行号，若省略则为第一行；ALL 表示传递全部变量，若省略，传递变量由 COMMON 语句定义；“DELETE 范围”表示程序调入后，删除指定范围内的程序行。

COMMON <变量名表>

功能:定义在执行 CHAIN 语句时需传递的变量。其中，“变量名表”中可以是变量、字符串和数组，当是数组时，在数组名后面加“()”。COMMON 语句应放在 CHAIN 语句前面，当 CHAIN 选用了 ALL 选择项时，COMMON 语句可以省略。

上述语句的使用方法可分以下两种情况进行讨论。

(1) 当一段程序执行完后，立刻调入下一段程序，直到全部结束为止。

例：设有三个程序分别为：A.BAS、B.BAS 和 C.BAS 各自程序如下：

A.BAS　　　　B.BAS　　　　C.BAS

```
10 REM A                 10 REM B                  5   REM C
20 ...                   20 ...                    10 ...
50 CHAIN B, 10, ALL      40 CHAIN C, 5, ALL        100 END
```

程序的执行过程：键入 MBASIC A 后，A 程序开始运行，运行列第 50 行时，调入程序 B 并运行，当程序 B 运行到第 40 行时，调入程序 C 并运行，直到第 100 句运行全部结束。

(2) 程序 A 在运行中调入程序 B，程序 B 执行完后返回到程序 A 继续执行.

例：设有二个程序，分别为 A.BAS 和 B.BAS 各程序如下：

```
A.BAS                    B.BAS
10 REM A                 10 REM B.BAS
20 ...                   20 ...
50 CHAIN B, 10, ALL      500 CHAIN A, 60, ALL
60 ...
100 END
```

程序执行过程为：当程序 A 执行到第 50 行时调入程序 B，B 程序执行列第 500 行时返回程序 A，并从第 60 行继续执行。

227. 如何在输入数据时伴有乐音

数据录入员在数据量输入时，由于眼睛要频繁交替的盯着屏幕和文稿，这样不仅眼睛容易疲劳，数据也容易出错。如果在输入时能用听力来辨别输入的对与错，就会大大降低眼睛的负担，提高录入速度。在 BASIC−80 中利用 BEEP 语句就能达到这一目的。

语句格式：BEEP <音调>，<时间>

其中，音调的范围从 0 到 255，0 为最高音、255 为最低音，时间的范围也在 0 至 255 之间，255 约为 1 秒种。

例：输入的数据是从 0 到 9 之间的数字，程序如下：

```
10 DIM A(100)
20 FOR I=1 TO 100
30 INPUT A(I)
40 IF A(I)<0  OR A(I)>9 THEN  30
50 BEEP A(I)*25, 100
60 NEXT I
70 END
```

其中第 40 句为:当发现数据出错时重送这一个数；第 50 句 A(I)*25 为了把音调拉开，这样容易辨别。

在刚开始输入数据时，由于对乐音与数字的关系不熟悉，可能效果不明显，但当输入一段时间后，就会熟悉了。到那时，录入员的眼睛不用看屏幕，靠听力就可以辨别出数据的对、错了。

228. 在CP／M系统方式下如何打印图形

虽然GBASIC语言作图功能很强，但由于缺少图形打印手段，使作图功能的应用在一定程度上受到了限制。为解决这一问题，谨向读者提供一种方法，以供参考。

(1)编程打印图形原理

若要编程打印图形，必须了解中华学习作图的基本原理和作图区地址与图象的关系。GBASIC作图的原理与APPLESOFT方式下的作图原理基本上相同，只需说名以下几点：

a.根据CP／M系统与监控程序地址的映射关系，作图区第一页应该从地址1000H开始，第二页从3000H开始。

b.屏幕显示图形时是横向按地址扫描，而打印机在打印图形时，8列点为一组纵向打印，一个单元内容要分7次才能打完。为了得到打印图形与显示图形统一的效果，在打印前必须分别把显示区8行单元的同一位与其它位拆开，并组成一个新的数据送往打印机。如下图所示：

```
1000H   B01 B02 B03 B04 B05 B06 B07
1200H   B11 B12 B13 B14 B15 B16 B17
1400H   B21 B22 B23 B24 B25 B26 B27
1600H    .                       .
1800H    .
1A00H    .                       .
1C00H    .                       .
1E00H   B71 B72 B73 B74 B75 B76 B77
```

按位拆开后送往打印机的图形数据如下图所示。

```
B01 B11 B21 B31 B41 B51 B61 B71
B02 B12 B22 B32 B42 B52 B62 B72
B03 B12 B23 B33 B43 B53 B63 B73
 .                           .
 .                           .
 .

B07 B17 B27 B37 B47 B57 B67 B77
```

若用高级语言来编写打印图形程序，由于拆字时间很慢，所以打印时间就很长。若不拆字而直接打印，打印的图形与屏幕相比正好旋转了90度。

c.若直接用打印语句输出，在连续打印了256个数据后会自动插入一些空格，这无疑会破坏图形的完整形象。为此需用下列语句完成打印功能。

1040 IF PEEK &HE1C1 < &H80 THEN POKE &HE091, DATA ELSE 行号 n;
其中 DATA 为图形数据。

(2) 打印图形程序

下面是打印图形的实例，本程序没有采拆字的办法，所以使图形旋转了 90。

例 1：按 1：1 的方式打印图形。

```
1000 BUF &H1000:ESC $ ="CHR $ (27)"
1005 LPRINT ESC $;"3";"!"
1010 FOR I=39 TO 0 STEP -1:A1=BUF+I
1015 LPRINT SPC(20);"K";CHR $ (192), CHR $ (0)
1020 FOR J=0 TO 80 STEP 40 :A2=A1+J
1025 FOR K=0 TO 916 STEP 128 :A3=A2+K
1030 FOR M=0 TO 7168 STEP 1024:A4=A3+M
1035 DAT=PEEK(A4) AND &H7F
1040 IF PEEK(&HE1C1)<&H80 THEN POKE &HE090, DAT ELSE 1040
1045 NEXT M, K, J
1050 LPRINT CHR $ (&H0D);CHR $ (&H0A)
1055 NEXT I
1060 END
```

若要打印黑底白线的图形；只要将 1035 行改为：

1035 DATA=(255-PEEK(A4)) AND &H7F 即可。

例 2:按 2:1 的比例打印图形。

```
10 D0=&H1000:LPRINT CHR $ (27);"3";CHR $ (21)
20 FOR D1=39 TO 0 STEP -1 :D=D0+D1
30 LPRINT SPC(23);CHR $ (27);"K";CHR $ (128);CHR $ (1);
40 FOR D2=0 TO 80 STEP 40 :D6=D+D2
50 FOR D3=0 TO 896 STEP 128 :D7=D+D3
60 FOR D4=0 TO 7168 STEP 1024:D5=D7+D4
70 DD=PEEK(D5) AND &H7F
80 FD=0:IF DD=0 THEN 120
90 IF DD>=64 THEN FD=48:DD=DD-64
100 IF DD>=32 THEN FD=FD+12:DD=DD-32
110 IF DD>=16 THEN FD=FD+3:DD=DD-16
120 IF PEEK(&HE1C1)<&H80 THEN POKE &HE090, F0 ELSE 120
130 FD=0 :IF DD=0 THEN 90
140 IF DD>=8 THEN FD=192:DD=DD-8
150 IF DD>=4 THEN FD=FD+48:DD=DD-4
160 IF DD>=2 THEN FD=FD+12:DD=DD-2
170 IF DD>=1 THEN FD=FD+3
180 IF PEEK(&HE1C1)<&H80 THEN POKE &HE090, FD ELSE 180
```

```
190 NEXT D4, D3, D2:LPRINT" "
200 NEXT D1
210 END
```

229. CP／M 系统中如何对文件进行保护

文件保护是一项经常性的工作，可以防止在操作失误时文件丢失。保护文件的方法有很多种，下面向读者介绍几种简单的保护方法:

(1)用 STAT 命令保护文件。

STAT 命令有改变文件属性的功能，通过文件属性的改变，可达到对文件保护的作用。

命令格式 1: STAT ＜文件名＞ $R／O

功能：把指定文件的属性改成只读型。用户只能对文件进行显示、打印和执行，而不能够修改、写入和删除。运用此功能可防止误操作造成的文件丢失。

命令格式 2: STAT ＜文件名＞ $SYS

功能：把指定文件的属性改成系统型。其特点是文件不能被显示、修改和删除，DIR 命令不能列出文件的名称，只能够运行文件。这样如同把文件隐藏了起来，从而起到文件保护做用。

当要解除上述设置，可用下列命令:

命令格式 3:

STAT ＜文件名＞ $DIR

功能：解除文件的系统属性，该命令是命令格式 2 的逆操作。

命令格式 4:

STAT ＜文件名＞ $R／W

功能：把文件属性置为可读可写型，它是命令格式 1 的逆操作。

上述命令的文件名均可以使用替代符“＊”和“?”。这样可对一批文件进行操作。

(2)对 BASIC−80 程序加密。

BASIC−80 系统本身具有程序加密功能，其命令为:

SAVE“文件名”，P

用该命令存盘的程序不能被 LIST、LLIST 等命令显示和打印，从而对文件有一定的保护作用。

230. 如何不关机实现冷启动

冷启动指计算机打开电源后，操作系统被载入和对系统初始化的过程。由于某种原因，使系统不能正常工作，热启动(CTRL−C)又失效时，冷启动可使系统迅速脱离非正常工作状态。

通过程序，我们也可以在不关机的情况下实现“冷启机”，这样可减少开关机的次数，延长机器使用寿命。

在 CP／M 系统中，为用户提供了 BOOT.COM 程序。该程序即可对 13 扇区的磁盘进行引导，也使能引导 16 扇区的磁盘，使用时，只要键入：BOOT 并根据提示选择扇区数后即可完成冷启动操作。

如果读者手上没有 BOOT.COM 程序，也可自己建立一个 BOOT 程序，方法如下：

在 CP／M 系统下键入：

```
A>DEBUG
-S100
0100 0E 01 CD 05 00 21 77 C7 20 00 30 21 00 C6 22 D0 F3 2A DE F3 33 00 30 .
-^C
A>SAVE 2 BOOT.COM
```

以后只要须要时，键入:BOOT 计算机就会自动进行“冷启动”。

231. 在 CP／M 系统中如何实现分页打印

分页打印是指打印的文件具有一定的格式，如:每页有相同的行数并有页号。这样，文件不仅美观而且便于装订保存和阅读。CP／M 系统没有直接提供此项功能，为此向读者推荐一个程序。通过它能够实现对各种文本文件进行分页打印。

程序清单如下：

```
5  HOME
10 INPUT "ENTER FILE NAME:";FN$
15 INPUT "ENTER LINES/PAGE: ";LP
20 I=1:ON ERROR GOTO 80
25 OPEN "I", #1, FN$
30 LPRINT "***";FN$;"***";SPAC(40);"PAGE(";I;")"
35 LS=LP
40 WHILE LS>0 AND NOT EOF(1)
45        LINE INPUT#1, A$
50        LPRINT A$
55      LS=LS-1
60 WEND
65 LPRINT CHR$(12):I=I+1
70 IF NOT EOF(1) THEN GOTO 30
75 CLOSE #1:GOTO 85
80 PRINT "FILE IS NOT FOUND!"
85 SYSTEM
90 END
```

使用方法：键入:RUN，屏幕显示：

ENTER FILE NAME：文件名　　输入要打印的文件名

ENTER LINES／ PAGE: 行数　　输入每页打印行数

随后，打开打印机就可以分页把程序打印出来。

232. 如何在一张盘上建立多个目录

在一般情况下，CP／M 系统只有一个目录，所有文件均存于该目录下。当目录中的文件较多，用户在检索时就会感到不方便。如果在一张盘上有若干个目录，用户就可以把文件分门别类地存在不同的目录里，这样不仅可以加快文件检索速度，对文件也有一定的保护作用。CP／M 系统的 USER 命令就能够完成上述功能。

命令格式:USER <n> 0⩽n⩽15

功能:该命令可把目录分成 16 个用户区，平时系统默认为 0 区，即 USER 0。若想换到 1 区，只要键入 USER 1 即可。在不同用户区的文件可以同名，系统不会相互覆盖。系统命令只能针对当前用户区进行操作，而不影响其它区的文件。由于 USER 命令是内部命令，使用时也很方便。

233. 当键盘的个别键出现故障怎么办

计算机经过长期使用后，键盘经常出现一些故障.如:个别连发、击键后无反应等.并且，出问题的键大都是较常用的键，如不及时修理定会影响工作.出现问题的原因大多数是由于硬件故障引起的.在维修力不足或无时间维修时，可用修改软件的方发帮助解决.其原理及方法如下:

CP／M 操作系统中，有六个单元(0F3ACH—0F3B1H)专门用来弥补键盘小，符号不全的弱点.六个单元分为三组，每组第一个字节为某键的 ASCII 码.第二个字节为改变后的 ASCII 码.例如:第一个字节为 41H，第二字节为 31H 则说明 A 键的 ASCII 码 41H 改成了 31H，这样，A 键就变成了″1″键.

目前，系统定义的内容为:

地址	内容	字符	说明
0F3ACH	0BH	CTRL—K	用 CTRL—K 键代替[键
0F3ADH	5BH	[	
0F3AEH	00H	CTRL—@	用 CTRL—@键代替 RUB(清除键)
0F3AFH	7FH	RUB	
0F3B0H	02H	CTRL—B	用 CTRL—B 键代替＼键
0F3B1H	5CH	＼	

系统定义的这些键其实并不常用，所以，改变其中的内容不会影响系统的正常工作.

例:用″%″号代替″A″键.

修改方法:开机后，键入:DDT (或 DEBUG)，系统进入 DDT 状态后顺序键入:

```
A>DDT↓
-SF3AC↓  0B 25↓
  F3AC    5B 41↓
  F3AE    00 .↓
-^C
A>
```

通过上述修改，再按"%"键时，屏幕将不在显示"%"，而显示"A"了.

当然，本方法并没有从根本上解决问题.只有排除硬件故障，问题才能得到根本的解决.

234. 如何在开机后自动执行用户程序

开机后自动运行用户程序是指计算机"冷启动"时除将操作系统装入内存外，还把用户程序装入内存并开始运行。此项功能可使程序使用更加简单，和自动化。

能实现此功能的程序是：AUTORUN.COM，该程序的使用方法是：开机后，键入：

A>AUTORUN <命令>

其中，<命令>指：可以立刻执行的程序名、系统的内部命令或外部命令。操作完成后，系统若再冷启动就会把<命令>的内容送入命令缓冲区并执行。

例：开机后自动执行 BASIC 程序 LPAGE.BAS。

A>AUTORUN MBASIC LPAGE

若要撤销此功能，只要键入

A>AUTORUN 即可。

235. 如何在单驱动器上复制文件

用一个驱动器复制文件可归纳成下列三种情况：

(1) 复制所有文件。此项操作可直接用 COPY 命令来完成。复制前先要对目标盘进行格式化，否则复制失败。

命令格式：COPY A:=A: 屏幕显示：

```
APPLE II CP/M

16 SECTOR DISK COPY PROGRAM
(C) 1980 MICROSOFT
INSERT MASTER DISK AND PRESS RETURN
```

这时插入原盘并按回车键，驱动器转动后屏幕显示：

INSERT SLAVE DISK AND PRESS RETURN

这时再插入目标盘并按回车键。然后重复以上两步直到显示：

COPY COMPLETE DO YOU WISH TO MAKE ANOTHER COPY?

为止，再键入"N"复制工作结束。

(2) 复制部分文件到另一张盘上。此项操作必由 FMT.COM 文件完成。

命令格式:FMT <目标文件名> = <原文件名>

其中文件名可以使用代替符“＊”和“?，这样一次可复制多个文件。当输入命令后屏幕显示的提示内容与 COPY 命令基本相同。

(3) 在同一张盘上复制文件。此项操作可以用 PIP.COM 文件完成。

命令格式: PIP <目标文件名> = <原文件名>

此时目标文件名不能与原文件名同名。若利用 FMT 也能完成此操作，只要把原盘与目标盘看做一个盘即可。

例:①把所有 BASIC 文件拷贝到另一张磁盘上。

A>FMT ＊.＊ = ＊.BAS

②用文件 OLD.COM 复制一个 NEW.COM。

A>PIP NEW.COM = OLD.COM

236. 在 CP / M 系统中如何恢复误删除的文件

在计算机的使用过程中，由于一时操作失误，造成文件丢失的情况是时有发生的，这无疑会给程序员代来烦恼和不必要的损失。为减少一旦事故发生时带来的损失，向读者介绍一种恢复被 ERA 命令删除文件的方法，以供参考。

CP／M 操作系统判断文件是否存在，是通过检测磁盘文件目录的属性位来断定的。磁盘目录分布在磁盘第三道的第 0、3、5、6、9、12、14 和 15 这 8 个扇区中。每个扇区能够存放 8 个文件目录信息，这样，一张盘最多可放 64 个文件。一个文件目录信息占 16 个字节，各字节的函意定义入下：

第 1 个字节为文件属性域，其值为 00H—10H 和 E5H，00H 到 10H 表示户号，初值均设为 00H。当文件被删除时，此位为"E5H"

第 2 到 10 个字节为文件名域，用来存放文件名。

第 11 到 12 没用。

第 13 到 16 为指针域，用来存放文件指针。

系统执行删除命令时，把删除标记"E5H"送入属性域，而不改变其它内容，只要不存入新的文件，被删除文件的全部信息始终被保留。我们只要利用这一点，通过某种方法把属性域的"E5H"改成"00H"，被删除的文件就可得到恢复。恢复的方法如下：

首先准备一个扇区编辑软件，在 COPY4.3 和 COPY5.0 中都有。这里以 COPY4.3 为例。

(1) 将 COPY 4.3 盘插入磁盘驱动器，开机调入 COPY Ⅱ PLUS 4.3，用“→”键移动光标在功能“菜单”中选择 SECTOR EDITOR 功能，按“回车”键，再取出 COPY 4.3 盘并

将要恢复文件的盘插入驱动器按。

(2) 进入扇区编辑后，键入:R(读扇区命令)，然后输入磁道号 03 和扇区号，由于事先不知道文件存在哪个目录扇区，所以先从 0 号扇区开始，若没找到再从 3、5、6…等扇区中查找。

(3) 当找到要恢复的文件名后，把光标移动到文件名的前一个字节(属性域)，这时该位内容应为“E5H”。

(4) 键入:H、00，把当前单元改成“00”。

(5) 键入:W，把修改后的内容存盘。

(6) 键入:ESC 键，退出编辑程序。

(7) 磁盘改好后重新开机引导。键入 DIR 命令，屏幕上会重新列出被删除的文件名，这时还应该对文件进行一次读写操作。文件如果是 BASIC 程序，则先进入 MBASIC，用 LOAD 命令取出，再用 SAVE 命令存入；若是其它文本文件，则用 ED 程序对文件进行读写。当完成了上述步骤后，文件就被恢复了。

237. 如何使用行编辑程序 ED

ED 程序是专门用来输入，修改文本文件的编辑程序，具有较强的行编辑功能，是程序员输入文件的重要工具。

ED 程序的工作过程简述如下:

当 ED 对指定文件 (源文件)进行编辑时，先将源文件装入内存缓冲区，然后在磁盘上建立一个临时文件，扩展名为:“. $ $ $ ”，用户通过键盘和屏幕对缓冲区的程序进行编辑修改，修改结果可通过子命令送入临进文件保存，当编辑结束进时，ED 把源文件改名为后备文件，扩展名为:.BAK，又把缓冲区所有内容存入临时文件，再把临时文件名换成原文件名，然后退出编辑。在编辑过程中，ED 有四个指针，它们是：原文件指针 SP，内存缓冲区文件指针 MP，临时文件指针 TP 和字符指针 CP。它们分别指示各文件当前处理的位置，如:当源程序没被装入内存时 SP 指向第一行，当程序全部装入内存缓冲区后 SP 指源程序最底部；而 MP 指向缓冲区文件的最末行； TP 指向临时文件的末行；CP 指示缓冲区文件的当前字符。

ED 程序有如下子命令:

nA: 从原文件中调入 n 行到内存缓冲区。

±B: 把 CP 指针移到缓冲区文件的顶或底部。

±nC: 从 CP 开始，向前或向后移动 n 位。

±nD: 从 CP 开始，删除 n 个字符。

E: 结束编辑，将缓冲区文件存盘并退出。

nF<串>: 寻找字符串。

H: 结束编辑，将缓冲区文件存盘，但不退出。

I: 在 CP 处插入字符。

nJ: 搜索、取代及删除。

±nK 从 CP 处删除 n 行。

±nL　CP 向上或向下移动 n 行。

nM　宏命令可以同时把多个命令放入一行。

nN<串>：找寻字符串并自动扫描整个文件。

O：　回到原来的文件。

±nP：输入从 CP 开始的第 n 页。

Q：　放弃修改，回到 CP／M 系统。

R：　读入文件库。

nS<串>：替代字符。

±nT：从 CP 开始，显示出 ±n 行。

±U：+U 输入时，把小写字母改为大写，−U 输入时字母不做变动。

nW：把缓冲区的内容存入临时文件。

±nX：把 n 行程序传送到库文件中。

±nZ：暂停。

±n↓　将 CP 从当前位置移动 ±n 行。

n：　将 CP 移到第 n 行.

在使用 ED 前，应检查一下磁盘的可用空间是否够用，通常应留出至少为二倍文件的空间，否则当编辑结束时由于磁盘空间不够，会造成写文件失败，使整个编辑工作前功尽弃。

238. 如何使用 PIP 程序

PIP 程序是输入输出设备间文件传送程序。其传送方式有：磁盘对磁盘的传送;磁盘对打印机的传送；磁盘对屏幕的传送；屏幕对磁盘、打印机的传送，还可以与穿孔机进行数据交换。在传送过程中能对文件进行处理，以满足不同设备和不同用户的需要，该程序还为用户提供了扩充功能的接口，使用户可以把自编的驱动程序装入 PIP 。

PIP 有两种工作方式，一种为命令方式，一种为交互方式，命令方式一次只能执行一种操作，执行完后立刻退出；交互方式则在系统进入 PIP 状态后，可多次执行不同的传送命令，直到用户要求退出时为止。PIP 工作状态提示符为“ * ”号。

命令格式:

(1) 命令方式：PIP <命令串>

(2) 交互方式：PIP

* <命令串>

.

.

.

*

其中<命令串>表示 PIP 的二级命令，格式及功能如下：

[d:]fn1 · ext = [d:]fn2 · ext

功能：把文件 fn2 · ext 拷贝到 fn1 · ext，其中 d: 可以是不同的磁盘驱动器。

d1: = d2:fn · ext

功能：把 d2: 中的 fn · ext 拷贝到 d1: 中，且文件名相同。

d:fn · ext=d1:fn1 · ext，d2:fn2 · ext…。

功能：把 d1:与 d2: 中的文件连接起来，拷贝到 d:的 fn · ext 文件中。

DEV: = fn · ext [，fn2 · ext…]

功能：把 fn1 · ext 文件传送到 DEV: 设备上，若选任选项为 fn1 · ext 与 fn2 · ext，则连接后再送到 DEV: 设备上。DEV: 可表示下列设备：

LST: 打印机，可完成文件打印功能。

CON: 显示器，可完成文件显示功能。

PRN: 打印机，可印出文件的行号和页号。

fn · ext = DEV:

功能：把 DEV: 设备的文件送往 fn · ext 文件中。其中 DEV: 可表示下列设备：

CON: 把屏幕上的内容送入文件 fn · ext。

DEV1: = DEV2:

功能：把 DEV2: 设备的文件送到 DEV1:。

其中 DEV1: 可表示：LST:、PRN:、CON:、A:、B:等。
此外 DEV1: 还可表示：OUT:，该设备是用户自定义的输出设备，设备驱动程序由用户自编，PIP 在执行时，自动转入 0106H 单元。在 0106H、0107H 和 0108H 单元中用户事先应设置一条 JP 指令以指向驱动程序的入口地址并把要输出的字符送到 C 寄存器中。DVE2: 可表示为：INP: 该设备为用户自定义的输入设备。PIP 在执行时转入执行 0103H、0104H、0105H 单无的 JP 指令，转入用户驱动程序，返回时从 0109 单元取回输入信息。

在 PIP 的命令中还可以加带一些命令参数，PIP 在执行命令时，可根据参数对文件进行处理。参数需放在命令行未，并用“[ ]”号括起来。该括号可由 CTRL–K 和 CTRL–M 键产生。常用参数有：

E：把传送的信息显示在屏幕上。

Gn: 把第 n 个用户区的文件拷贝到当前区。

L：在传送时把大写字母变成小写字母。

N：在传送过程中为文件加入行号。

O：在传送时，不把文件结束标志 CTRL–Z 做为文件结束，而由目录中给出的文件大小来决定。

Pn: 分页打印，每页 n 行，省略 n 时为 60。

R：可拷贝系统文件。

S<串 1>CTRL–Z，Q<串 2>CTRL–Z: 可完成从串 1 到串 2 之间的部分文件传送，串 1 和串 2 均为文件中的信息。

239. 如何使用 STAT 程序

STAT.COM 程序在 CP／M 系统中能够向用户提供有关磁盘文件的信息、磁盘的信

息、系统设备配置的信息，还能通过 STAT 命令对系统配置进行修改，是一个非常实用的程序。下面分别介绍 STAT 程序的功能与使用方法。

1. 了解磁盘文件信息

STAT 程序可通过下列命令了解文件信息：

命令格式：STAT d:<文件名>

文件名若用 *·* 代替则显示全部文件信息。如：

A> STAT *·*

REC	BYTES	EXT	ACC	
13	2K	1	R/W	A:APDOS·COM
64	8K	1	R/O	A:ASM·COM
8	1K	1	R/O	COPY·COM

BYTES REMAINING ON A:118K

其中 REC 为记录数，1 个记录为 128 个字节，BYTES 为字节数，单位为"K"；EXT 表示文件占用了多少 16K 空间；ACC 表示文件的属性，R/W 表示可读可写，R/O 表示只读，SYS 表示系统型等；A: 为驱动器名；最的字符为磁盘文件名；最后一行显示出磁盘所剩的可用空间。

2. 了解磁盘信息

命令格式 1：STAT

屏幕显示：A:R/W，SPACE:118K

其中 A: 表示 A 盘，R/W 表示 A 盘为可读可写盘，SPACE: 118K 表示可用空间。

命令格式 2：STAT d:DSK

屏幕显示：

d:Drive Characteristics	d: 为指定驱动器。
1024: 128 Byte Record Capacity	共 1024 个记录，每个记录为 128 个字节。
128: Kilobytb Drive Capacity/	总容量为 128 K。
48: 32 Byte Direetvry Entries	48 个目录项，每项 32 个字节。
48: Checked Directory Entries	48 个可检查目录项。
128: Records / Exfent	每个范围 128 个记录。
8: Records / Block	每个块 8 个记录。
32: Seetors / Tracks	每个磁道 32 个扇区。
3: Reserved Tracks	保留了 3 个磁道供系统使用。

3. 了解设备信息

命令格式：STAT DEV:

CP/M 系统规定了 6 种物理设备和 4 种逻辑设备其名称为：

物理设备：

TTY: 电传打字机

CRT: 显示器

BAT: 批处理理机，如：读卡机、打印机等

PTR: 纸带输入机

LPT:　　打印机

PTP:　　纸代穿孔机

逻辑设备:

CON:　　控制台

RDR:　　读卡机

RUR:　　穿孔机

LST:　　列表机

系统在初始化时已将物理设备对应到了相应的逻辑设备上，上述命令可显示出它们的对应关系。如: A>STAT DEV: 屏幕显示:

CON:　IS　CRT:　　CRT　做为控制台。

RDR:　IS　RTR:　　纸带输入机做为读卡机。

PUN:　IS　PTP　　纸带穿孔机做为穿孔机。

LST:　IS　LPT　　打印机做为列表机。

4. 重新定义外部设备

命令格式: STAT 逻辑设备: =物理设备:

例如:　STAT LST:=CRT:表示把列表机定义成显示器，这样，不管以后使用 CTRL-P 或 LLIST 都指把文件显示在屏幕上。

5. 改变文体属性

命令格式 STAT <文件名> $S

其中 S 可以是: R/W、R/O、SYS、DIR 使用方法请参阅第 229 问。

6. 命令提示

当命令格式不清楚时可使用提示命令。

命令格式: STAT VAL:键入该命令后屏幕上会显示出所有 STAT 命令的格式，以供参考。

九、dBASEⅡ部分

240. 如何在dBASEⅡ中使用数组

dBASEⅡ的数据类型中缺少数组类型，这里向读者介绍两种模拟数组的方法。

1.用内存变量模拟数组：

用做数组的内存变量名由两部分组成，前一部分为字母，可作为数组名，后一部分为数字，可视为下标。如A1，A11等。在使用时先要对数字部分进行处理，得到确定的下标值，再把其值转化成字符串与前一部分字母连接形成一个完整的内存变量名，然后把它存放到另一个内存变量里，最后对这个变量进行宏代换操作就可得到一个数组元素值。

例：输入10个数并显示在屏幕上。

```
SET TALK OFF
STORE 0 TO I
DO WHILE I<10
        STORE I+1 TO I
        STORE 'A'+STR(I,2) TO M
        INPUT 'ENTER DATA: " TO &M
ENDDO
LIST MEMORY
RETRUN
```

鉴于dBASEⅡ的规定，内存变量又不超过64个。所以，上述方法只适于数据量较少的情况。

2.用数据库文件模拟数组功能。

(1) 模拟一维数组：

首先建立一个只有一个字段的数据库，把字段名看作数组名，记录号做为下标。使用时，可用GOTO n、SKIP n等定位命令控制记录指针就能够取到所要的数据。

(2) 模拟二维数组：

若把记录号做为行下标，字段名做为列下标，这样数据库就可看成二维数组。记录号的确定参考:(1)；列下标的设定参考:用内存量模拟数组的方法，只要把内存变量改为字段名变量就可以了。

例：对5＊5的矩阵进行转置。设矩阵数据已存于数据库文件D.DBF中。该库结构如下：

字段名	类型	长度	小数位
A1	N	5	0
A2	N	5	0

```
A3        N        5        0
A4        N        5        0

SET TALK OFF
USE D
LIST OFF
STORE 0 TO I
DO WHILE I<=4
        STORE I+1 TO I
        STORE I TO J
        DO WHILE J<=5
                GO I
                STORE 'A'+STR(J,1) TO AR1
                STORE &AR1 TO VAL1
                GO J

                STORE 'A'+STR(I,1) TO AR2
                STORE &AR2 TO VAL2
                REPLACE &AR2 WITH VAL2
                STORE J+1 TO J
        ENDDO
ENDDO
LIST OFF
RETURN
```

使用本方法应注意数组“下标”的行不能正超过 65535 列不能超过 32 列。

如果把数据库文件名也看做下标，还能够实现模拟三维数组。这里就不再举例了。

241. 如何自动修改数据库结构

程序运行时经常要用临时文件来存放计算过程中的中间结果。在 dBASEⅡ中临时文件的生成一般事先通过 CREATE 命令建立好文件结构，程序运行后再向里面追加数据。这样做的结果不但使文件的数量增加，使磁盘空间的开销也随之增加，在临时文件较多的情况下更是如此。若用 COPY STRUCTURE 命令也可产生临时文件结构，但产生的结构与原来的相同，使用起来很不方便，若再用 MODIFY STRUCTRE 命令对结构进行修改又会破坏程序运行的独立性。为克服在使用临时文件时采用上述命令带来的缺点可试用下述方法。

本方法的基础设想是:利用原有的库结构，对其进行自动修改，然后再生成新的库结构。原有的库若简称“原库”，要建的临时库简称“新库”，则修改步骤为:

(1) 用 COPY TO <文件名> STRUCTURE EXTE 命令把原库的结构存入结构描述库。结构描述库的结构由系统自动生成，一共有 4 个字段，内容及含意如下：

字段名	类型	长度	小数位
FIELD:NAME	C	010	0
FIELD:TYPE	C	001	0
FIELD:LEN	N	003	0
FIELD:DEC	N	003	0

字段:FIELD:NAME、FIELD:TYPE、FIELD:LEN 和 FIELD:DEC 分别存放原库的字段名、类型、长度和小数点位数。原库中描述字段的信息在描述库中表示一条记录，原库有多少字段，描述库中就有多少条记录。

(2) 用 REPLACE <字段名> WITH <表达式>命令对描述库的记录进行修改，若修改 FIELD:NAME 字段相当于修改原库的字段名，若修改 FIELD.TYPE 相当于修改原库的类型，以此类推。

(3) 用 CREATE <文件名> FROM <结构描述库>命令使结构描述库的记录转化为新库的结构。

通过以上步骤就可完成自动修改库结构的过程。

例:统计学生成绩库 G 中每班各科的总分。G 库的字段名含意为:C:NO 班号、S:NO 学号、MATH 数学成绩，PHYS 物理成绩。CHEM 化学成绩。

程序如下：

```
SET TALK OFF
USE G
COPY STRUCTURE TO SG EXTE
USE SG
GOTO 3
REPLACE NEXT 3 FIELD:LEN WITH FIELD:LEN+2
CREATE MG FROM SG
USE MG
APPEND FROM G
TOTAL ON C:NO TO OUT
UST OUT
LIST
USE
DELETE FILE SG.DBF
DELETE FILE MG.DBF
DELETE FILE OUT.DBF
RETURN
```

程序第三至八行完成了对 MATH、PHYS 和 CHEM 字段的修改，使它们的宽度比

原来增加了 2 个字节，这样可以有效的防止执行 TOTAL 命令时统计数据溢出。

242. 如何增加内存变量的数量

在 dBASEⅡ 中最多只允许使用 64 个内存变量或容纳 1536 个字节的数据，这对有些程序是远不够用的，为此我们可采用分段存盘的办法，使问题得到缓解。

在一个程序中当内存变量使用到一定数量，而还要继续增加时，可先用 SAVE <文件名>命令把当前的内存变量存盘保存起来，然后用 RELEASE 命令释放这些变量所占有的内存空间，这样就可以增加新的变量了，当需要使用原来的变量时，再将当前变量存盘，然后用 RESTORE FROM <文件名>命令将上一次存盘的变量恢复到存贮器中，这样就能够继续使用了。

```
set talk off
STORE 1 TO A
STORE 'Y' TO B
SAVE TO M1
RELEASE
STORE 2 TO A
STORE 'N' TO B
SAVE TO M2
RESTORE FROM M1
? A,B
RETURN
```

243. 如何巧用排序命令

排序命令 SORT 和索引命令 INDEX 在程序中会经常用到，它们可根据关键字段的数值或 ASCII 码值的大小对记录进行排序。但在实际工作中，常常会遇到要求按职称、职务、姓氏笔划等与 ASCII 码顺序无关的内容排序的情况。此类问题若直接使用 SORT 命令或 INDEX 命令是无法完成的。为此向读者介绍一种解决此类问题的方法，以供参考。

当要排序的数据库记录较少时可采取增加排序码字段的方法。即对所有记录，在原来的基础上增加一个字段存放排序码。排序码可根据不同对象来设定，如对职务进行排序时职务高的排序码大，职务低的排序码小。这样对职务的排序就可转化为对排序码进行排序了。这种方法简单易行，但因增加了排序码字段，使整个库的体积有所增加，同时增加了数据输入量，所以当记录较多时不易采用。

当记录较多时可采用建立排序码库的方法，排序码库共有两个字段，分别用来存放排序码和职务名称，这样，该库记录数只与职称种类有关，所以数据量很小。当要对某库职称字段进行排序时，先以职务字段为关键字将该库与排序码连接，使之增加排序码字段，然后对排序码字段进行排序，排序后再将该字段删除，这样就完成了排序工作。

例:对职称进行排序，排序码库为 SC.BDF;被排序的库名为 FIL,DBF 排序码库 SC.DBF 的结构及记录:

字段名	类型	长度	小数位
A	C	3	0
B	C	10	0

A	B
1	实验员
2	助理工程师
3	工程师
4	高级工程师

被排序库的结构:

字段名	类型	长度	小数位
B	C	20	0
C	C	5	0
.	.	.	.
.	.	.	.

```
SET TALK OFF
SELE SECO
USE SC
SELE PRIM
USE FIL
JOIN ON B TO FIL1 FOR B=S.B
USE FIL1
SORT ON A TO FIL2
USE FIL2
COPY TO FIL3 FIELD B,C
USE FIL2
DELE FILE FIL1.DBS
LIST
RETURN
```

用上述方法同样能完成对汉字笔划的排序，这里就不详述了。

244. 如何巧用宏代换函数&

宏代换函数&是一个功能很强的函数，巧妙的使用它能使程序设计的更加灵活和简练。

宏代换的基本功能如下:

1. 可以替代常量和变量;
2. 可以替代文件名;

3. 可以替代表达式;

4. 可以替代命令;

下面向读者介绍一例用宏代换函数替代条件判断的方法。

在科技成果管理系统库中，一般要专设一字段，用来注明科研成果通过鉴定的级别，如:国家级、部门级、省市级等。为减少输入信息量和节约外存空间，在建库时一般只输入各级别的代码如 A、B、C 等。设:A 代表国家级、B 代表省市级、C 代表部门级。在输出时再把代码转换成级别名称，供用户查看。

用宏代换完成此项功能的程序如下;

假定成果系统库中，通过鉴定级别的字段名为 LEVEL，则程序清单如下:

```
SET TALK OFF
STORE '国家级' TO A
STORE '部门级' TO B
STORE '省市级' TO C
* SF 为成果库
USE SF
DO WHILE .NOT. EOF
        STORE LEVEL TO QU
        ? &QU
        SKIP
ENDDO
USE
RETURN
```

通过第八句的宏代换，直接完成了将代码 A、B、C 转化成"国家级"、"部门级"、"省市级"的工作，省去了条件判断，使程序更加简练。

245. 如何实现 dBASEⅡ与 BASIC-80 之间的数据传送

dBASEⅡ除了能对数据库进行有效的管理外，还能把数据库文件转化成系统数据文件供在其它语言环境下使用，还能把系统数据文件转化成数据库文件。这样就使得 dBASEⅡ在使用中更加灵活。如当数据计算量很大时把它们转化成系统文件，用 BASIC-80、FORTRAN-80 等语言来处理，当系统数据文件修改量较大时可转为数据库文件，由 dBASEⅡ来完成，这样可以充分发挥各自的优势，提高运行效率。

在数据库中能完成上述功能的命令格式为:

COPY TO ＜文件名＞ SDF 〔条件〕

其功能是:把当前的数库文件按一定条件拷贝成系统数据文件。

APPEND FROM ＜文件名＞ SDF 〔条件〕

其功能为:把指定的系统数据文件追加到当前数据库中。

下面是能完成上述功能的程序:

```
USE DF                DF 为数据库名
```

```
COPY TO SM SDF
QUIT 'B:MBASIC BP'          BP 为 BASIC 程序名

5  REM BP
10 OPEN "R" #1,"A:SM.TXT",24
20 FIELD 10 AS N$,4 AS M$,4 AS C$,4 AS T$, 2 AS CF$
30 FOR CODE%=1 TO 10
40 GET #1,CODE%
50 YY%=CVI(M$)+CVI(C$)+CVI(T$):T$=MKI(YY%)
60 PUT #1,CODE%
70 NEXT CODZ%
80 CLOSE #1
90 SYSTEM

USE DF
DELETE ALL
PACK
APPEND FROM SY SDF
LIST
RETURN
```

在数据库中，DF 库的结构为：

字段名	类型	长度	小数位
NAME	C	10	0
SX	N	4	0
YW	N	4	0
YY	N	4	0

在 BASIC 程序的第 20 句中，多说明了一个字段:CF$，这是为读取每行记录后面的"回车"、"换行"这两个数据而设置的，若忽略了这一点，读入数据就会出错。

246. 如何解决数据库字段不够用的问题

在 dBASEⅡ中规定:数据库字段不能超过 32 个，但在实际应用中常常会遇到 32 个字段不够用的情况。如:在人事档案管理中就会遇到这类问题。通常的解决办法有两种。一种是把多个字段压缩成为一个字段；另一种是把一个大库拆成两个小库使用，下面分别加以介绍:

1.多字段压缩:

当字段较多时，可将两项或多项内容以字符串的形式存放到一个字段里，使用时再用

命令分解出来。例如在人事档案中把姓名和年龄送入一个字段的程序如下：

```
USE DR                          DR为人事档案库
STORE ″Y″ TO L
DO WHILE L=″Y″
        ACCEPT ″NAME:″ TO A
        ACCEPT ″AGE :″ TO B
        REPLACE AB WITH $(A,1,10)+$(B,1,3)
        WAIT TO L
ENDDO
RETURN
```

当须要查询时用下列程序：

```
USE DR
DO WHILE .NOT. EOF
        STORE AB TO C
        STORE $(C,1,10) TO A
        STORE $(C,1,3) TO B
        ? ″NAME ″,A
        ? ″AGE  ″,B
ENDDO
RETURN
```

2. 拆成两个或多个库：

当32个字段不够用时，可以把它们一分为二存放到两个库中。这两个库要有相同的关键字，如:″编号″，并在该字段上建立索引文件。然后按例题中给出的方法处理就可以将两个库当做一个库使用了。

例:把档案信息分别存于EXP1.DBF和EXP2.DBF两个库中，它们的结构分别如下：

EXP1.DBF结构

FLD	NAME	TYPE	WIDTH	DEC
001	A	N	2	0
002	B	C	2	0
003	C	C	2	0
004	D	C	2	0

EXP2.DBF结构

FLD	NAME	TYPE	WIDTH	DEC
001	E	N	2	0
002	F	C	2	0
003	G	C	2	0

程序：

```
SELE SECE
USE EXP2
SELE PRIM
USE EXP1

SET LINGAGE ON
```

DISPLAY ALL FIELD P.A,B,C,D,S.E,F,G FOR P.A=S.E

在运行上述程序时，使用两个库与同使用一个库的效果一样。

247. 在dBASEⅡ中如何调用汇编程序

在dBASEⅡ V2.3B版本和V2.4版本中增加了调用汇编程序的功能，其命令格式为：

1.POKE <addr>，<code〔,list〕>

<addr>:为存贮器地址。

<code>:为代码，或代码表。

功能:从指定地址开始，装入代码，当代码多于一个时，用逗号将其分隔。地址和代码均为十进制数。

2.SET CALL TO <addr>

<addr>:汇编程序口地址。

功能:为CALL命令设置入口地址。

3.CALL〔val〕

〔val〕:参数，入口时送往HL寄存器。

功能:转入设定的入口地址，执行汇编程序，执行完后，返回主程序。

调用汇编的基本步骤是:

用POKE命令将汇编的机器码送入存贮器;再用SET CALL TO命令设定程序入口地址;然后用CALL命令调用子程序。

```
POKE 29377,195,0,0
SET CALL TO 29377
CALL
```

执行该程序会退出dBASEⅡ，返回CP/M系统。

248. 如何避免用@命令打印时出现的走纸现象

@命令是可用来显示或打印的命令，常用的命令格式为:

@ X,Y SAY <表达式>

其中X,Y是用来定位的座标;<表达式>是要显示的内容。

由于该命令有座标定位功能，所以在输出表格时常被采用。但一些用户在用做打印机输出时忽略了Y的取值范围，所以在打印中常出现走纸现象。造成该现象的原因是由于Y的值超过了256，所以只要不使Y值超过256就可避免走纸现象的发生。

249. 如何在打印机上印出制表线

使用dBASEⅡ离不开打印表格，表格打印的好坏关系到程序使用的效果。

在中华学习机上，如果使用汉字 dBASEⅡ可以直接使用汉字库中的制表符，这些制表符有:“、、〈、」、″、、【、、′、、、】等，在区位码输入方式下输入对应的区位码就能够把它们调出来。它们的区位码分别是:0916、0948、0928、0922、0964、0941、0924、0956、0928、0904、0906。

如果设有汉字 dBASEⅡ，利用打印机内部提供的制表符也能印出满意的表格来。由于苹果机配置的打印机型号各不相同，制表符的 ASCII 码有很大差异，所以，使用时要根据实际情况而定。这里给出几种常用打印机制表符的 ASCII 码:

START-80		PCA-80	
符号	ASCII 码	符号	ASCII 码
┌	C9	┌	B8
└	C8	└	BA
┐	BF	┐	B9
┘	D9	┘	BB
┬	CB	┬	B1
├	C7	├	B3
┤	B9	┤	B2
┼	C5	┼	AF
─	CD	─	B5
│	BA	│	B6

下面以 PCA-80 为例说明在程序中的使用方法。程序清单如下:

```
SET TALK OFF
STORE 0 TO I
STORE '     ' TO K
STORE 'CHR(181)+CHR(181)+CHR(181)+CHR(181)' TO X
STORE 'CHR(179)+&X+CHR(179)+&X+CHR(175)+&X+CHR(178)' TO Z
STORE 'CHR(183)+K+CHR(183)+K+CHR(183)+K+CHR(183)' TO H
SET PRINT ON
? CHR(184)+&X+CHR(177)+&X+CHR(177)+&X+CHR(185)
DO WHILE I=5
          STORE I+1 TO I
          ? &H
          ? &Z
ENDDO
? &H
? CHR(186)+X+CHR(176)+C+CHR(176)+X+CHR(187)
SET PRINT OFF
RETURN
```

250. 如何用好 BROWSE 命令

BROWSE 命令是具有对数据库文件进行全屏幕编辑功能的命令，灵活，合理的使用该命令能够大大提高对数据文件修改的速度。但由于 BROWSE 命令要求屏幕宽度为 80 列，所以若在只能显示 40 列的中华学习机上使用就会出现一些问题。如当显示的数据库记录长度大于 40 个字符时屏幕会出现数据复盖现象。这在一定程度上妨碍了该命令的使用。

其实，如能充分了解该命令的全部功能，复盖现象是能够避免的。在 dBASEⅡ2.4 版本中 BROWSE 命令的格式为：

BBROWSE〔FIELDS<字段名集>〕〔FOR<表达式>〕

利用可选项〔FIELDS<字段名集>〕可以限制显示字段的数量，从而达到减少每行字符数的目的，不仅如此，该选择项还屏蔽了不需要修改的字段。使修改的速度和可靠性得到提高。但是当某个字段本身的长度超过 40 个字符，该可选项就无能为力了，这时只能用缩短字段长度的办法来加以解决。

251. 如何在输出打印时使零变为空格

在编辑表格程序时，常常希望当输出数据为零时不把零打印出来，而用空格来代替零的位置，这样，不仅打印出来的表格整洁美观，重点突出，而且还减少了打印机头的磨损。这里向读者介绍一种实用的零变“空”的方法。

在数据输出前，先把所有要输出的数据转移到一个全部字段类型为 C 型组成的数据库文件中，该库的结构可以预先建好，也可参考第 241 问的方法临时建立。数据转移完毕后，将字段所有为零的数据改为空格，这样在输出打印时就不会出现零了。

例:输出打印工资清单。

工资数据库名为 GZ.DBF，其结构为：

FLD	NAME	TYPE	WIDTH	DEC
001	BH	C	3	0
002	XM	C	10	0
003	JBG2	N	6	2
004	BT	N	6	2
005	BSJ	N	6	2
006	SFGZ	N	6	2

程序清单：

```
SET TALK OFF
USE G
REPLACE ALL SFGZ WITH JBGZ+BT-BSJ
COPY TO SG STRU EXTE
COPY TO GT SDF
```

```
USE SG
REPLACE ALL FIELD:TYPE WITH 'C'
CREATE OUT FROM SG
USE OUT
APPEND FROM GT SDF
REPLACE ALL BT WITH ' ' FOR BT='0'
REPLACE ALL BDJ WITH ' ' FOR BSJ='0'
SET PRINT ON
LIST
SET PRINT OFF
DELETE FILE GT.TXT
DELE FILE SG.DBF
RETURN
```

程序的4–10行用来建立OUT.DBF文件，该库所有字段均为C型。第11–12行用来将所有为零的数据改为空格，13行以后为输出打印。

252. dBASEⅡ2.4版本与2.3版本的主要区别是什么

dBASEⅡ2.4版本继承了2.3版的全部优点并在功能上有所加强，它们的主要区别为:

一、系统文件数量减少

原2.3版本共有12个系统文件，运行时系统调入调出次数频繁，影响运行速度。而2.4版本把系统程序压缩成四个，使磁盘读写次数有所减少。

二、增加命令

2.4版本增加的命令有:

1.HELP命令

功能:可显示dBASEⅡ命令和函数的使用说明，帮助用户学习使用dBASEⅡ。

2.REINDEX命令

功能:对当前已建索引文件的数据库文件在经过修改后可按原来的关键字和文件名重新建立索引文件。

3.DISPLAY／LIST STATUS命令

功能:显示系统当前的状况，如:已经打开了几个文件，它们所在的工作区，当前工作区是主区还是次区，SET命令设置情况，默认驱动器为A还是B等。可帮助用户了解系统当前的情况。

4.TEXT–ENDTEXT

功能:键入TEXT后向屏幕输入任何信息，而不会被当做命令。当键入ENDTEXT后该功能撤销。

三.新增加的函数:

1.RANK(<字符串>)函数

功能:给出字符串首字符的ASCII码。

2.TEST(<表达式>)函数。

功能:测试表达式的类型并得到一个函数值，当函数值为1时表达式为逻辑型。值小于1时表达式为字符型。当值大于1时，表达式的值为字符型。

四.对有些命令的功能进行了扩充。

1.APPEND. INSERT. EDIT 命令。

当 SET FORMAT TO <格式文件>命令生效后，屏幕显示方法由格式文件决定，并由 READ 命令配合使用，这样使屏幕的显示更加灵活。

2.BROWSE 命令

该命令增加了可选项〔FIELDS<字段名表>〕这样在做浏览时可不必显示全部字段，从而加快了浏览速度。

3.REPLACE 命令

该命令增加了可选项〔NOUPDATE〕以避免对索引文件进行修改。

4.SAVE 命令

该命令增加了两个可选项:〔ALL LIKE<通配符>〕和〔ALL EXCEPT<通配符>〕。通配符为“*”或“?”，*表示一串字符。?代表一个字符。若选择前一可选项，在存贮内存变量时，只存通配符所表示的变量。例如：通配符为 VAL* 则表示存贮 VAL1、VAL2、VAL10 等。若选择后一可选项，则表示只存贮不包括通配符所表示的那些内存变量。

5.RESTORE 命令

谈命令增加了可选项 [ADDITIVE] 意为在恢复内存变量时不破坏当前的内存变量。

6.UPDATE 命令

该命令增加了可选项 [RANDOM]，选择该项后使指定的修改文件不用排序。

十、PASCAL 语言部分

253. APPLE–PASCAL 系统介绍

APPLE–PASCAL 系统实际上是 U.C.S.D–PASCAL 操作系统，也叫 P–SYSTEM。P(Pseudo)是伪的意思。

该系统的主要设计思想是把原 CPU 通过软件改造成“理想”的 CPU 或叫做 P–机器。它有一套独立于原 CPU 的指令代码系统，称 P–代码。系统在运行程序时先由编译器把原程序编译成 P 代码文件。然后再由解释程序把 P 代码文件翻译成原机器代码来执行。由于系统采用了这种结构，使它具备了如下特点。

1. 可移植性。由于整个系统在 P–机器上运行，所以与原 CPU 的环境无关，这样不管是何种机型，只要具备 UCSD–SYSTEM 操作系统、并且有相同标准的存贮介质、它们生成的 P–代码无需做任何修改就可相互移植。

2. PASCAL 语言的整体性。

P 系统的指令绝大部分是由 PASCAL 语言写成，这使得 PASCAL 语言不但是一种高级语言，而且是操作系统的开发语言和其他高级语言的编译语言，如 FORTRAN77。从而进一步扩充了 PASCAL 语言的能力，使操作系统与 PASCAL 语言配合的更加密切。

3. 适合于在内存容量小的环境下运行。

由于 P 系统结构化程序高和解释性的操作方式，使它能够在较小的内存容量下运行。

APPLE – PASCAL 系统实际上是苹果机上的 P–SYSTEM 。

为充分利用苹果机的工作环境，在硬件配置上 APPLE–PASCAL 系统定义了 11 个输入输出设备。它们的代号、名称如下所示。

设备号	设备名称	用途
#1:	CONSOLE:	控制台(指屏幕和键盘)
#2:	SYSTEM:	系统
#3:	1 号盘	做为引导系统的驱动器
#4:	2 号盘	存放系统文件和程序
#5:	PRINTER:	输出打印机.
#6:	REMIN:	远程输入，用于通讯
#7:	REMOUT:	远程输出，用于通讯
#8:	5 号盘	存贮数据
#9:	6 号盘	同上
#10:	3 号盘	同上
#11:	4 号盘	同上

上述设备均固定的按排在苹果机的 8 个槽口上。具体分配如下:

槽口号	设备接口卡名称	用途
0	16K RAM 卡	存放解释程序和 I／O 系统
1	打印机卡	连接 PRINTER
2	通讯接口卡	连接 REMIN／REMOUT
3	串行口卡	用以连接外部终端
4	驱动器卡	连接#9 和#10 驱动器
5	同上	连接#11 和#12 驱动器
6	同上	连接#4 和#5 驱动器

中华学习机是苹果机的兼容机，所以 APPLE-PASCAL 同样也可以在中华学习机上运行，只是在硬件环境设置上有所不同。

其一，16K 卡，驱动器卡已和主板合为一体所以无需再用此卡。

其二，目前中华学习机只能接一台磁盘驱动器，做为 1 号盘，设备号为#4，如想增加驱动器，需做一些改进，请参考第 310 问。

其三，中华学习机只有一个扩展槽，一般用做 1 号槽，插打印机卡，设备号为#6。

目前 APPLE-PASCAL 系统可支持的语言有：PASCAL、FORTRAN77、6502 汇编。此外还有一些应用软件和表演程序。对于 PASCAL 系统，所有软件分别存放在四张盘上，各盘的名称分别为：APPLE0、APPLE1、APPLE2、APPLE3，这四张盘上各存的文件如下：

APPLE0:
SYSTEM.PASCAL
SYSTEM.MISCINFO
SYSTEM.COMPILER
SYSTEM.EDITOR
SYSTEM.FILER
SYSTEM.LIBRARY
SYSTEM.CHARSET
SYSTEM.SYNTAX

APPLE1.
SYSEM.APPE
SYSTEM.PASCAL
SYSTEM.MISICINFO
SYSTEM.EDITOR
SYSTEM.FILER
SYSTEM.LIBRARY
SYSTEM.CHARSET
SYSTEM.SYNTAX

APPLE2.
SYSTEM.COMPILER
SYSTEM.LINKER
SYSTEM.ASSEMBLER
6502.OPCODES
6502.ERRORS

APPLE3.
SYSTEM.APPLE
FORMATTER.CODE
FORMATTER.DATA
LIBRARY.CODE
LIBMAP.CODE
SETUP.CODE
BINDER.CODE
CALC.CODE
LINEFEED.TEXT
LINEFEED.CODE
SOROCGOTO.TEXT
SOROCGOTO.CODE
SOROC.MISCINFO
HAZELGOTO.TEXT

HAZELGOTO.CODE
DISKIO.CODE
HAZEL.MISCINFO
DISKIO.TEXT
CROSSERF.TEXT
CROSSREF.CODE
SPIRODEMO.TEXT
SPIRODEMO.CODE
HILBERT.TEXT
HILBERT.CODE
GRAFDEMO.TEXT
GRAFDEMO.CODE
GRAFCHARS.TEXT
GRAFCHARS.CODE
TREE.TEXT
TREE.CODE
BALANCED.TEXT
TREE.CODE
BALANCED.TEXT
BALANCED.CODE

APPLE–PASCAL 系统命令具有良好的层次结构，不管进入那一层，该层的命令均在屏幕的第一行显示出来以供选用。最高层为 COMMAND 层，该层共有 11 个命令，它们的功能如下：

F(ILE：	调用文件管理程序；
E(DIT：	调用文本编辑程序,输入或修改文件
C(OMPILE：	调用编译程序,对原程序进行编译
A(SSEMBLE：	调用6502汇编编译程序
L(INK：	调用连接程序,把外部的P–代码程序与当前的PASCAL程序连接
X(ECUTE：	装入P–代码程序并运行
R(UN：	编译并运行当前工作文件
D(EBUG：	动态调试程序,目前尚未实现
U(ESR RESTART：	重复上一次的命令
I(INILIALIZE	使系统初始化
H(ALT	使系统重新启动

此外，系统还设定了 5 个功能键，它们是：

CTRL–A　屏幕换页。APPLE–PASCAL系统的屏幕宽为80列，但苹果机的屏幕只有40列，为能在一行内容纳80个字符，系统把屏幕分为两页，每页 40 列，这样就能满足系统要求了。

CTRL–Z　页面跟踪。该功能使屏幕页面随光标左右移动而移动。

CTRL–@　程序中断。中断后按空格键，系统重新初始化。

CTRL–F　禁止输出命令。当按此键后程序禁止输出，但还继续运行。

CTRL–S　暂停命令。当按此键程序暂停运行，直到再按键后才继续运行。

总而言之，APPLE–PASCAL 系统是一个功能很强操作系统，它能够帮助用户更好的发挥苹果机的潜力，成为用户的得力助手。

254. APPLE–PASCAL 系统命令与系统文件的对应关系

APPLE–PASCAL 系统的命令层共有 10 个命令，执行每一个命令都必须具备该命令所需在的系统文件和运行环境，否则命令将无法执行。下面给出命令与文件的对应关系：

命令	需要的系统文件	文件位置及说明
F(ile	SYSTEM.FILER	任何驱动器，只在调用时才装入内存。
E(DIT	SYSTEM.EDITOR	同上
C(OMPILE	SYSTEM.COMPILE	同上
	SYSTEM.LIBRIRY	必须在引导盘上。
	SYSTEM.EDITOR	当编译出错后，按 E 键后调入内存。
	SYSTEM.SYNTAX	在任何盘，当使用编译任选项 S 时调入
A(ssemble	SYSTEM.ASSEMBLE	任何磁盘
	6520.OPCODE	同上
	6520.ERRORS	同上
	SYSTEM.EDITOR	同上
L(ink	SYSTEM.LINKER	同上
X(ecute	可执行 P–代码文件	同上
	SYSTEM.CHARSET	在任何盘，当在绘图方式中用列 IOCHAR、WSTRING 语句时被调用
R(un	SYSTEM.WRK.CODE 或 SYSTEM.WRK.TXT	在引导盘上。
	SYSTEM.COMPILER	若无 SYSTEM.WRK.CODE 时被调入
	SYSTEM.EDITOR	若编译译出错时被调入
	SYSTEM.SYNTAX	当用到编译任选项 S+时被调入
	SYSTEM.LINKER	当需要连接多个子程序时被调入
	SYSTEM.LIBRARY	必须在引导盘上
	SYSTEM.PASCAL	同上
	SYSTEM.CHIRSET	同上
U(ser restart		
I(nitialige	SYSTEM.PASCAL	必须在引导盘中
	SYSTEM.MISCINFO	
H(alt	SYSTEM.PASCAL	
	SYSTEM.APPLE	
	SYSTEM.MISCINFO	

了解了上述关系后，在调试时就应当把要用到的系统文件装到相应的磁盘上，以确保系统正确运行。

255. 函数与系统库的对应关系

为能在较小的内存空间里运行大程序，APPLE–PASCAL 系统将大量标准函数集中在系统程序库(SYSTEM.LIBRARY)的子库中，只有在使用 USES 语句调用时子库时才被装入内存。所以在使用函数时必须了解它们是属于哪一个子程序库。在系统库中，共包含 4 个子库，它们与函数的对应关系如下:

1、 TURTLEGRAPHICS　海龟作图子程序库。该库包含的函数过程有：

INITTURT	进入高分辨率作图模式；
TEXTMODE	进入字符显示模式；
GRAFMODE	转入作图屏幕；
VIEWPORT(L.R.B.T)	建立作图窗口；
FILLSCREEN(C)	为作图框涂色；
PENCOLOR(C)	定义画笔颜色；
MOVETO(X.Y)	画笔移至 X,Y 处；
MOVE(D)	画笔移动；
PURN(A)	逆时针转动画笔；
TURVTO(A)	确定画笔方向；
TURTLEX	给出画笔的 X 坐标；
TURTLEY	给出画笔的 Y 坐标；
TURTEANG	给出画笔的方位角；
SCREENBIT(X,Y)	判断指定坐标处屏幕颜色是否为黑；
UCHIR(C)	在图形方式下显示字符变量 C；
WSTRING(S)	在图形方式下显示字符串变量 S；
CHARTYPE(n)	在图形方式下确定字符的颜色与低色；
PRAWBLOCK()	数组作图。

2、 APPLESTUFF　子程序库。该库包含的函数过程有：

RANDOM	随机数过程；
RANDOMIZE	改变 RANDOM 函数的初始值
KEYPRESS	按任意键,该函数值为:TURE
NOTE(X,Y)	音乐函数；
PADDLE(n)	游戏杆控制函数；
BUTTON(n)	游戏控制器按钮控制函数
TTLOUT(n,b)	向第 n 个输出器输送电平，b 为真时为高电平，否则为低；

3、 TRANSCEND　超越函数子库。该库包含下列函数：

SIN(X)	正弦；COS(X)　余弦；EXP(X)　指数；
SQRT(X)	开方；LN(X)　对数；LON(X)　以 10 为底的对数；
ATAN(X)	反正切。

4、 CHAINSTUFF　子程序库。该库包含下列函数：

SETCHAIN(文件名)	执行指定的 P–代码文件
SETCVAL(字符串)	把字符串存入系统保存。
GETCVAL(字符串变量)	把在系统中保存的变量取回。
SWAPON	内存复盖；

在编程时，若用到某个子程序库，必须在程序名语句后面加 USES 语句来调用。

语句格式：USES <子库名称> [，<子库名称>…];

```
例    PROGRAM    P10.3
      USES TRANSCEN0;
      .
      .
```

若用到多个子库时，库名与库名之间用逗号分开。

256. 如何制作适合中华学习机的系统盘

APPLE-PASCAL 的系统盘共有四张，为适应不同配置的用户，每张盘的程序都做了精心安排，如：APPLE0 较适合单驱动器用户，APPLE1 和 APPLE2 适于多驱动器的用户。但上述安排并非最佳，特别是在单驱动器方式下使用，要经常更换磁盘，这对上机操作十分不便。中华学习机只能装一个驱动器，所以有一个使用方便的单驱动器系统，对于中华学习机更是由为重要。

在 APPLE0 中有下列文件

SYSTEM.TASCAL、　　SYSTEM.MISCINFC
SYSTEM.COMPILER、　　SYSTEM.EDITOR
SYSTEM.FILER、　　SYSTEM.L113RARY
SYSTEM.CHARSET　　SYSTEM.SYNTAX

分析以上文件不难看出，经常换盘的主要原因是缺少 SYSTEM.APPLE 文件，使该磁盘无引导能力，所以，应把 SYSTEM.APPLE 文件装入 APPLE0，但由于 SYSTEM.APPLE 占用了 32 个磁盘块，而 APPLE0 中只剩有 28 个磁盘块，无法装入。因此必须删除一些文件腾出一些空间。参考第 254 问的内容，不难找出不常用的文件。如：SYSTEM.SYNTAX、SYSTEM.CHARSET 等。进入文件管理层 F(ILE，用 R 命令把它们删除，再用 T 命令把 SYSTEM.APPLE 拷贝到 APPLE0 中这样，APPLE0 即能引导，也能运行程序了。整理后的 APPLE0 有下列文件：

SYSTEM.APPLE　32　　SYSTEM.PASCAL　41
SYSTEM.MISCINFO　1　　SYSTEM.COMPILER　75
SYSTEM.EDITOR　47　　SYSTEM.LIBRIRY　34

所剩磁盘空间为 16 块，能满一般小程序的运行，若还想增加一些空间，可参阅第 266 问，把 SYSTEM.LIBRARY 文件加以修改，去掉不用的函数缩小该库所占空间。留出更多的空间为用户使用。这样 APPLE0 的剩余空间可接近原来的平水。

上述修改，最好不要在原盘上进行。可先做一次拷贝，然后直在拷贝盘上进行修改，以防操作失误后无法恢复。

257. 如何在 APPLE-PASCAL 系统下绘图

图形功能是 APPLE-PASCAL 系统的一大特点。

作图是在屏幕上进行的，其清晰度为：192 行×280 列；屏幕左下角的坐标为(0,0)、右上角的坐标为(279,191)。图形屏幕与文本屏幕的转换方法通常有两种，一种方法是，调用无参过程 GRAFMODE 使屏幕由字符模式转换到图形模式。另一种方法是通过 TURTLEGR 过程来转换。从图形方式切换成文本方式一般由 TEXTMODE 过程来实现。

APPLE-PASCAL 的作图方法有三种，分别是：海龟作图、坐标作图和图形数组作图。

海龟作图是利用命令控制画笔(海龟)向前、或向左、右转弯，在画笔经过的地方会留下一条线。在使用海龟作图前必须调用系统库中的 TURTLEGR 子库，调用方法是:在变量说明语句前加入 USES TURTLEGR 语句。

画笔控制语句如下:

TURNTO(角度):用来控制画笔转向，角度在-359 至 359 之间，若超出范围则为 360 的余数。

TURN(角度增量):将角度增加一个给定的量。

MOVE(距离):将画笔从当前位置起沿原定方向移动一段距离，并画出线来。

此外，海龟作图还有四个函数

TURTLEX:给出画笔的 x 坐标。

TURTLEY:给出画笔的 y 坐标。

TURTLEANG:给出画笔的角度。

SCREENBIT(x,y):回送一个布尔量，若如 TRUE 说明 x,y 坐标处荧光屏不是黑色的，反之为黑色。

坐标作图法是给出屏幕坐标使画笔移动到该处并画一条线。语句格式为:

MOVETO(x,y)

画笔移动后，原“龟头”的角度不变。

图形数组作图是事先把图形画面存到一个压缩的二维数组中，数组元素为布尔型，为 TURE 时，表示白色，反之为黑色。作图时可把整个数组抄送到屏幕，也可在数组元素中开矩形窗口，把窗口内的元素抄往屏幕。此法可加快绘图速度。语句格式为:

DRAWRLOCK(A,B,X1,Y1,W,H,X,Y,N)

其中：A 为数组，B 为表示数组中每行所占字节数。其值可根据每行点数 P 来计算。公式是:

B = (P+15)DIV60 * 2。

X1,Y1 表示将数组中图形窗口送往屏幕时窗口左下角的起点坐标。W 为窗口宽度、H 为窗口高度；X、Y 表示数组送往屏幕的屏幕坐标;N 表示图形显示模式，值域为从 0 至 15，其含意为:

0 屏幕填充黑色；

2 数组与屏幕颜色的补色相“与”；

4 数组取反并与屏幕颜色相“与”；

6 数组与屏幕颜色“异或”；

8 数组与屏幕颜色“与”；

10 数组复制到屏幕；

12 屏幕替换；

14 数组与屏幕颜色“或”；

1 数组与屏幕颜色“或非”；

3 屏幕颜色取补色；

5 数组取反；

7 数组与屏幕颜色“与非”

9 数组与屏幕颜色相同；

10 数组复制到屏幕；

11 数组和屏幕颜色的补色“或”；

12 屏幕替换；

13 数组的“反”与屏幕颜色“或”；

15 将白色填充屏幕。

图形颜色的控制语句为：

PENCOLOR（颜色）。PASCAL 系统共有下列颜色：

WHITE 白 0

WHITE1 白 1、两点宽，与绿色、紫色一同使用

WHITE2 白 2、两点宽，与橙色、兰色一同使用

BLACK 黑

BLACK1 黑 1、两点宽，与绿色、紫色一同使用

BLACK2 黑 2、两点宽，与橙色、兰色一同使用

GREEN 绿

VIOLET 紫

ORANGE 橙

BLUE 兰

其中，若只用黑白两色，可直接使用:WHITE、BLACK。若还用其它颜色时仍用:WHILE、BLACK 会使线条粗细不匀，故在有绿色和紫色时白色和黑色应用:WHITE1 和 BLACK1。

在着色方面，还有下列函数。

FILSCREEN (颜色)：该函数可使窗口内填满一种颜色。

REVERSE： 使颜色变为补色。原色与补色的关系如下所示：

原色：	WHITE	WHITE1	WHITE2	GREEN	ORANGE
补色：	BLACK	BLACK1	BLACK2	VIOLET	BLUE

反之亦然。

例：画出不同颜色的正多边形。

```
PROGRAM P259
USES TURTLEGR;
TYPE XC=0..279;YC=0..191;TN=2..8;LH=5..100;
PROCEDURE POLY(N:TN;SIDE:LH;X:XC;Y:YC);
VAR ANGLE:0..360;I:TN;
BEGIN
        ANGLE:=360 DIV N;PENCOLOR(NONE);
        MOVETO(X,Y);PENCOLOR(WHITE);TURNTTO(0);
        FOR I:= 2 TO N DO
```

```
            BEGIN
            MOVE(SIDE);TURN(ANGLE);
            END;
        MOVETO(X,Y);PENCOLOR(NONE);
END;
BEGIN (* MAIN *)
        INITTURTLE;POLY(3,50,2,20)
        POLY(4,40,50,60);POLY(5,32,96,100);
        POLY(6,30,155,135);POLY(8,50,190,10);
        MOVETO(0,182);WSTRWG('<RET>FOR ENDING!');
        READLN
END.
```

258. APPLE-PASCAL 系统如何在图形中显示字符

为能在图形方式下显示字符，APPLE-PASCAL 为用户提供了 SYSTEM · CHARSET 文件，在该文件中存贮了字符集的所有图形，使用起来非常方便。

在图形方式下显示字符的函数过程有:

WSTRING(字符串表达式):将字符串常量或变量印到画面上，每个字符为 7 点宽 8 高，首字符的左下角为图形方式下画笔的当前位置，当字符串显示完后画笔移至尾字符的右下角，当字符串超出屏幕宽度时，自动截去超出的字符。

WCHAR(字符型表达式):该函数只能印出单个字符，其显示方式同上。

CHARTYPE(方式):控制所显示字符的颜色。字符颜色不受 PENCOLOR( )函数影响，只与本函数的"方式"有关。方式的范围为:0 至 15，代表含意如下:

0	黑色底、黑色字	1	补色底、黑色字
2	黑色底、补色字	3	补色底、补色字
4	原色底、黑色字	5	白色底、黑色字
6	原色底、补色字	7	白色底、补色字
8	黑色底、原色字	9	补色底、字再与底补色
10	黑色底、白色字	11	补色底、白色字
12	原色底、白色字	13	白色底、原色字
14	原色底、白色字	15	白色底、白色字

例如:当屏幕是紫色时,"方式"=6，则显示的字为绿色。

例:

```
PROGRAM P260
USES TURTLEGR;
CONST D=14;H=12
VAR I,L,R,B,T:INTEGER;
MODE:STRING;COLOR:SCREENCOLOR
```

```
PROGEDURE COLORS;
BEGIN
      B:=0;T:=191;COLOR:=NONE;
      FOR I:=0 TO 19 DO
            BEGIN
            I:=I * D;R:=L+D-1;VIEWPORT(L,R,B,T);
            FILLSCREEN(COLOR);
            IF COLOR=WHITE2 THEN COLOR:=NONE
            ELSE COLOR:=SUCC(COLOR)
            END
END;
PROCEDURE MODES;
BEGIN
      FOR I:=0 TO 15 DO
            BEGIN
            MOVETO(0,I * H);CHARTYPE(10);
            STR(I,MODE);WSTRING(MODE);
            MOVETO(21,I * H);CHARTYPE(I);
            WSTRING('ABCD      1234      ')
            WSTRING('PRESS<RET> TO STOP')
            END
END;
BEGIN (* MAIN *)
      INITTURT;
      COLORS;
      VIEWPORT(0,279,0,191);
      MODES;
      READLN
END.
```

259. 在 APPLE-PASCAL 系统中如何演奏音乐

APPLE-PASCAL 有关演奏音乐的函过程只有一个格式:

NOTE(音调, 时间)

其中, 音调在 0 至 50 之间, 0 表示休止符, 2 至 48 为不同的音阶, 每增加 1 大约提高半个音阶, 音调为 20 时, 约为 C 调的 1。

时间指音调持续的时间, 数越大, 时间越长。数若增加一倍, 持续时间也延长一倍。

在使用该函数时, 必须先调用系统子库 APPLESTUFF。

例: PROGRAM P261;

```
USES APPLESTUFF;
VAR PITOH, OURATION: INTEGER
BEGIN
DURATION: =100;
FOR PITCH: =12 TO 20 DO
    NOTE(PITCH, DURATION)
END。
```

260. 在 APPLE-PASCAL 系统中如何使用打印机

使用打印机前，先将中华学习机扩展槽设成 1 号槽，并插上打印卡。使用打印机的主要目的是用来完成打印程序、打印运行结果和打印目录。打印原程序和打印目录的工作是在文件管理层 F(1le 内实现的。其方法是:

若打印原程序，在 F 层下键入“T”，屏幕显示:

TRANSFER WHAT FILE? SYSTEM.WRK.TEXT ↓ (文件名)

TO WHERE? #6:(或 PRINTER:) (设备号／名)

此时打印机若已打开就可把工作文件 SYSTEM.WRK.TEXT 打印出来。

若打印磁盘目录，在 F 层下键入 E 和#6，这时就可以打印出十分详细的磁盘目录。

在打印程序运行结果时可以把打印机看成一个输出文件，在进行文件说明时，文件类型可分别选为：TEXT、FILE OF CHAR 或 INTERASTIVE。

下面用一例题来说明用法。

例：打印出 ASCII 字符。

```
PROGRAM 262;
VAR P:TEXT                  或 P:FILE OF CHAR、P:INTERACTIVE
PROCEDURE LINE(A,B:CHAR);
  VAR CH:CHAR
  BEGIN
   FOR CH:=A TO B DO
     BEGIN
      WRITE(P,CH)
      WRITELN(P)


     END;
    END;
BEGIN(*MAIN)
      PRWRITE(P'PRINTER:')
      LINE(CHR(20),CHR(59));
      LINE(CHR(60),CHR(99));
      LINE(CHR(100),CHR(126));
```

```
CLOSE(P)
END。
```

程序第二行:P: TEXT 也可用 P: FILE OF CHAR 或 P: INTERACTIVE 来代替。

261. 如何打印 APPLE-PASCAL 系统绘制的图形

APPLE-PASCAL 系统没向用户提供图形打印的功能，若想要打印出绘制的图形，可以借助 COMPUTER STATIONS 公司提供的 DUMP 软件来完成。DUMP 软件的主要功能是把计算机图形区的内容传送到打印机。并能支持多种打印机接口卡，使用时有屏幕提示，非常方便。

打印 PASCAL 图形步骤如下:

用 PASCAL 程序绘出图形，完成后返回命令层，从驱动器中取出系统盘，并将 DUMP 盘插入，然后打开打印机并分别键入: CTRL-RESET、2、1、7 就可以完成打印了。

如果没有 DUMP 盘，在键入 CTRL-RESET 后，再重复一次，然后键入 PR#1、CTRL-Q，也能实现打印。

在打印前，不要忘记插打印卡。

262. 在 APPLE-PASCAL 系统中如何充分利用磁盘空间

当磁盘使用了一段时以后，会发生这样的现象: 明明磁盘上有足够的空间，确总显示磁盘满，使一些应该能够存贮的程序不能存入。

出现这种现象的原因是这样的: 系统在存文件时，先要在磁盘上查找一片连续的空间。在确认连续空间的总容量大于存入文件容量时，才能将该空间分配给文件。在磁盘经过多次读写删改后，会产生许多不连续的可用空间即是所谓的“零头”，虽然这些“零头”的总和很大，但每个“零头”确很小，所以无法利用。

系统为解决“零头”问题，在文件管理层 FILE 中提供了 K(runck 命令，该命令可使磁盘上的所有文件向顶部集中，让“零头”连续起来，这样就能够存贮文件了。

如: 当在做文件存贮或转存时，屏幕上显示出 OUTPUT FILE FULL 时，进入 F(ile 层，查看磁盘使入情况，键入 L,显示:

DIR LISTING OF ? #4

显示引导盘目录内容并在最后一行显示:

8／8 FILES 38 UNUSED ,36 IN LARGEST

其中 38 为可用空间的块数，36 为最大连续空间块数。当发现可用空间比要存贮的文件小时，说明文件确实存不下了，需要换盘或删除一些文件；当要存文件不大于可用空间时，说明磁盘零头太多，需要用 K 命令对盘进行整理；键入 K、#4、Y 系统就会对#4:驱动器上的文件进行移动，让零头连接起来。上述工作完成后，就可以进行文件存贮了。

263. 在APPLE–PASCAL系统中磁盘出现故障应如何解决

当磁盘经过长期使用后，很容易出现故障，造成读写失败。给用户工作带来很大的麻烦，为避免出现读写错误时使磁盘上的数据全部丢失，学会“修补”磁盘是很有必要的。

当出现读写错误时，先要判断一下是软盘的问题还是磁盘驱动器的问题，方法很简单，可另找一张盘进行读写操作试试，若仍出错说明是硬件的问题，需要检修驱动器；否则，说明是软盘有问题了。

当确定是软盘的故障后就可以进行修补了。

修补的方法如下：

在命令层键入 F、E；屏幕显示：

BAD BLOCKS SCAN OF ? #4:(#5:)

SCAN FOR 280 BLOCKS(Y／N) Y

列出坏磁盘块的清单：

BLOCK 10 IS BAD

BLOCK 11 IS BAD

BLOCK 12 IS BAD

4 BAD BLOCKS

键入 X 显示：

EXAMINE BLOCKS ON ? #4:(或 #5:) ↓

BLOCK–RANG 10–15 ↓

FILE(S) ENDANGERED

文件名 1.TXT 10 11

文件名 2.COD 12 13

FIX THEM ? Y

BLOCK 10 MAY BE OK

BLOCK 11 MAY BE OK

通过上述操作,坏块处的文件被移出,坏块加锁,即禁止对坏块进行读写操作.这时磁盘便可以继续使用了。在使用中还应注意,被移出的文件还可能有错误,应给与特别修改，如，对某程序重新编辑等。

264. 编译程序任选项的含意是什么、应如何使用

编译程序的任选项，是为原程序在编译过程中用来与编译程序交换信息以确定编译方式的字符串。其格式为：

(＊ $ <任选项>[, <任选项>…]＊)

在使用中应注意与注释语句的区别。注释语句的格式为：

（* <注释信息> *）

其中 $ 为任选项的前缀前面不能有空格。若选用多个任选项时任选项之间要用逗号隔开。任选项及其功能如下：

C：注解任选项。　C 后面的字符串可直接放入代码文件做为注释。

G+：GOTO 语句任选项。允许 GOTO 语句在程序中出现。

G：　不允许 GOTO 语句出现。

I+：I／O 检查任选项。　在 I／O 操作时产生 I／O 代码用以检查 I／O 是否正确。

I－：　不产生 I／O 代码。

I <文件名>：包含文件任选项。在编译中包含指定的源文件。

L+：表文件任选项。　在编译过程中产生列表文件，表文件名为 SYSTEM.LST.TEXT。

L－：　不产生列表文件

L <文件名>：　将编译列表送往指定文件。

N+："空载"任选项。　直到需要时程序包(UNIT)才被装入内存。

N－：　程序一运行程序包立刻被装入。

P：页任选项。　在列表文件中，在有 P 任选项的那一行插入一个换页符。

Q+：静默编译任选项。　在编译过程中"关闭"屏幕，不在显示编译过程中的信息。

Q－：　终止 Q+。

R+：区域检查任选项。　编译程序产生检查数组和字符串下标的代码，以及检查子域类型和串类型变量赋值代码。

R－：　终止 R+

R <程序包名>：驻留任选项。只要含有 R 任选项的过程或函数被调入存，不管它是否被 SEGMENT 语句说明，都将驻留在内存中。

S+：交换任选项。　在编译过程中，程序一段段的调入内存进行编译，这样可以编译更大的程序，但编译时间将延长一倍。

S－：　终止 S+

U+：用户名任选项。　通知编译程序编译用户程序

U－　编译系统程序。

U <文件名>：使用库任选项。指出要使用的库文件名，随后系统调用这个库中的程序函数。

265. 在 APPLE–PASCAL 系统中如何运行大程序

中华学习机的内存空间较小，只有 64K，为能利用这有限的空间运行更大的程序，APPLE–PASCAL 系统引入了内存覆盖技术，即把整个程序按需要分成若干段，程序用到哪段，哪段才被装入内存。装入时，先将前一段程序从内存中调出，再将后段程序装入。由于使用的都是同一个内存空间，所以使程序的大小与内存无关了。

分段方法如下：

在需要分段的函数或过程前面插入 SEGMENT。

例如：SEGMENT PROCEDURE SP

```
BEGIN
(* PASCAL 语句 *)
END;
SEGMENT PROCEDURE SF
BEGIN
(* PASCAL 语句 *)
END;
.
.
.
```

这里应特别注意分段的原则，因为怎样分段关系到各段调入调出次数是否频繁，若太频繁会严重影响程序的运行速度。一般说来，那些共用的过程或函数应常驻内存，而对使用次数较少的函数和过程可给与分段。

分段覆盖的功能可以通过编译选择项(* $R<过程或函数名> *)加以修改，使之能常驻内存，以满足某些场合的需要。如：过程 A 在运行时要多次调用函数 B，但 A 和 B 均已分段，都不驻留内存，当 A 被入时，立刻让 B 为常驻内存，这样就可以减少 B 被频繁调入的次数，从而提高程序的运行速度。

在分段使用时，还应注意：只要有一个 SEGMENT 过程将被调用，存有这个程序代码文件的磁盘就必须联机。此外 SEGMENT 过程必须是包含代码生成语句的第一个过程说明。

调试大程序，最好使用两个驱动器，建议读者参考第 310 问提供的方法对中华学习机加以改进，这对你调试程序会带来极大的方便。

266. 如何把过程装入系统库

APPLE-PASCAL 的系统库 SYSTEM.LIBRARY 是为用户提供多种功能的程序库，该库最大的特点是：调用简单；被调用的函数当程序运行时才被链接，大大宿短了程序的代码文件长度，为充分利用内存资源提供了极大的方便。不仅如此，该库还为用户继续扩充留有很大余地，可以让用户修改或把自己编写的过程和函数装入系统库，以便随时调用。下面介绍一下装入方法。

(1)系统库的内部结构

系统库内设有 0 至 15 号槽口(SLOT)，每个槽口可存放一个内部程序包(该包内可存多个过程和函数)，这样，系统库最多可存放 16 个内部程序包。目前系统库已占用的槽口为 0 至 6 号，因此用户最多还可建立 9 个内部程序包。

系统为了识别不同层次的程序，为各类程序规定了段号。段号的范围从 0 到 31，对各段号,系统做了如下规定：

0 段为系统段，1 段为主程序段，2 至 6 段为系统留用，7 至 15 段供规则程序包、外部过程、SEGMENT 过程和函数使用，16-31 段为内部程序包专用；其中，已占用的段号为：20、21、22、29、30、31，用户还可以使用除此以外的 11 个。

(2)建立内部程序包

内部程序包必须遵循一定的程序格式，格式如下：

```
(* $S+*)
UNIT <程序包名>; INTRINSIC CODE <段号 1>[DATA<段号 2>]
INTERFACE
PROCEDURE <过程名> (参数表);
IMPLEMENTATION
PROCEDURE <过程名>
     (* 过程说明 *)
     BEGIN
          (* 过程体 *)
     END;
BEGIN
END。
```

其中，(* $S+*)为编译任选项，见第 264 问。

程序包名，由用户自定；

<段号> 16 至 31 之间，末被占用的任何一个。

DATA　　本包所需要的数据文件，该文件也要分配段号。

INTRFACE　表示程序入口处。

IMPLEMENTATION　　表示过程从以下开始。

(3)程序包装入系统库

当程序按上述格式用编辑文件输入后，在命令层键入 C，对该文件进行编译，产生代码文件 SYSTEM · WRK · CODE。然后将 APPLE 3，装入#5 驱动器，在命令层分别键入 X、#5:LIBRARY。屏幕显示：

OUTPUT CODE FILE->SYSTEM.LIBRARY ↓

LINK CODE FILE -> SYSTEM.LIBRARY ↓

此时屏幕列出原系统库内所有内部程序包的名称、槽口号和段号：

0–(30)	LONGINTL	2452	8–	0
1–(31)	PASCALIO	1238	9–	0
2–(29)	TRANSCEN	1168	10–	0
3–(22)	APPLESTU	662	11–	0
4–(20)	TURTLEGR	5202	12–	0
5–(21)	TURTLEGR	386	13–	0
6–		0	14–	0
7–		0	15–	0

此时键入:=，系统将原库中的所有内容按原槽口号复制到新的系统库中，接着键入 N,显示：

LINK CODE FILE→SYSTEM · WRK · CODE

此时屏幕上又显示出了 SYSTEM.WRK.CODE 的段号、程序包名和槽口号，继续键入

1，即 1 号槽口显示：SLOT TO LINK INTO？ 6、即连入 6 号槽口。重复此过程，可将 SYSTEM · WRM · CODE 中所有段全部连入新库中。上述工作完成后，键入:Q 新系统库就安装完毕了。由于新库名与原库名相同，所以原库的内容被新库取代。最后还要对系统进行重新初始化，即在命令层键入 I，新的系统库就可以使用了。

(4)调用方法：

调用方法与系统标准程序包的调用方法完全相同。其格式为：

```
PROGRAM  <主件名>
USES     <程序包名>
(*说明部分*)
 BEGIN
        (*程序体*)
        <过程名>(参数表)
         .
         .
 END。
```

(5)例题：

内部程序包：

```
(*$S+*)
UNIT IP; INTRINSIC CODE 26
INTERFACE
PROCEDURE AD(VAR N:INTEGER);
IMPLEMENTATION
PROCEDURE AD
BEGIN
        N:=N+1;
END;
BEGIN
END.
```

主程序：

```
PROGRAM P268;
USES IP;
CONST COUNT=1;
VAR N:INTEGER;
BEGIN

FOR N:=COUNT TO 100 DO
        WRITELN('N=',AD(N));
END.
```

267. 如何建立用户程序包

APPLE−PASCAL 系统不但有系统程序库供用户使用，而且还允许用户建立自己的程序库。也叫规则程序包。它是常用函数和过程的集合，以代码文件的形式存于磁盘上，调用时，用连接程序把它与主程序连接起来。它的优点在于：磁盘上只有一个程序付本，而其它程序均可调用，这样节省了磁盘空间，另外，程序包可放在任何盘上，不象系统库那样，受必须放在引导盘上的限制。

1、建立程序包。

程序包必须遵循如下格式：

```
(* $S+*)
UNIT  <文件名>
INTERFACE
PROCEDURE  <过程名>(参数表)
IMPLEMENTATION
PROCEDURE  <过程名>;
     BEGIN
          (*PASIC 语句*)
     END;
BEGIN
END.
```

其中：文件名为：规则程序包的子库名称；

过程名为：过程名称；

参数表为：过程入口参数；

INTERFACE 为：软件包入口；

IMPLEMENTATION 为过程入口

输入方法如同其它 PASCAL 程序一样。

输入完后对其进行编译、转存，转存时文件名可任意。这样一次可建立多个子库程序。当子库建好后以后就可以建立用户程序包了。

建立程序包需调用 APPLE3：中的 LIBRARY · CODE 代码文件，具体方法是：

将 APPLE3 盘装入驱动器，在命令层建入 X、LIERARY

屏幕显示：

```
OUTPUT CODE FILE -> <程序包名>
LINK CODE FILE -> <子库代码文件名>
0-(7)  子库名  100     8-      0
1-              0      9-      0
2-              0      10-     0
3-              0      11-     0
4-              0      12-     0
5-              0      13-     0
```

6–	0	14–	0
7–	0	15–	0

此时有三种操作方式共选择：如建入 0↓表示要把 0 号槽的子库连接到程序包中。继续显示：

SLOT TO LINK INTO?

表示连接到程序包的几号槽中，可键入 0 至 15 的任意一个。如按 0。

表示连接到 0 号槽，接着键入 N，显示：

LINK COPE FILE→<子库代码文件名>

即输入第二个子库代码文件。重复上述操作，直到把所有程序连接到程序包中为止。最后键入 Q 退出。用户程序包至此建立完毕。

2. 调用方法

主程序在调用用户程序包时应按下列格式编写程序。

```
PROGRAM  <程序名>;
USES (* $U #4:<程序包名.CODE>) <子库名>;
        (* 说明部分 *)
BEGIN
        (* PASCAL 语句 *)
END.
```

当程序输入完毕、编译正确后还要进行连接。其方法是：在命令层键入 l。

显示：

```
APPLE PASCAL LINK [1,1]
HOST FILE ? SYSTEM WRK.CODE↓
OPENING SYSTEM.WRK.CODE
LIB FILE ? SYSTEM.LIBRARY↓
LIB FILE ? <用户库名>↓
MAD FILE ? ↓
OUTPUT FILE ? ↓
```

到此连接结束，系统回到到命令层。最后键入 R，程序开始运行。

3. 例题：

被链接程序

```
(* $S+ *)
UNIT LP;
INTERFACE
PROCEDURE P(VAR N: INTEGER);
IMPLEMENTATION
PROCEDURE P;
    BEGIN
        N:=N * N;
    END;
```

```
BEGIN
END.
主程序:
PROGRAM P269
USES (* $U #4:MYLLIBRARY.CODE *) LP
VAR I,N:INTEGER
BEGIN
    N:=2
    FOR I:=1 TO 10 DO
        BEGIN
        P(N);WRITELN('N=',N)
        END
END.
```

268. 如何在 APPLE-PASCAL 中建立开动系统

当程序调试完毕后，简化使用步骤是必须考虑的问题，建议读者把程序改为自动开动系统，使系统起动就自动执行用户程序，这样可以省去记忆文件名、和操作规程等诸多不便。修改方法如下。

(1). 将要用到的程序库装入引导盘

(2). 把程序的代码文件名改为 SYSTEN · STRTUR。

当引导盘空间较小可将一些系统文件如: SYSTEM · FILER. SYSTEM · EDITOR 或 SYSTEM · SYSTAX 拷贝到其它盘上，以腾出一些磁盘空间。

例: 将 MYDISK:盘上的工作文件改为开动文件。在命令层键入: F、C、SYSTEM · WRK · CODE、SYSTEM · STRTUR、Q、H。此时系统会重新启动，并自动运行用户程序。

十一、LOGO语言部分

269. 如何让LOGO演奏音乐

在LOGO语言中，没有专门的音乐命令，所以要想使LOGO能够演奏音乐，必须分两步进行。第一步：用过程把用6502机器语言编写的音乐程序指令代码送入不被系统占用的地址空间，如$0300。第二步：通过不断调中用该程序，把乐谱数据给音乐程序，从而实现让LOGO演奏音乐的功能。具体方法如下：

```
    TO  MC: DATA
    MAKE "ADDRESS  770
1:  IF: DATA=[ ]  STOP
    DEPOSIT  : ADDRESS  : DATA
    MAKE  "DATA  (BF  : DATA)
    GO  1
    END
```

执行该过程用如下命令：

?MC [173 48 192 136 208 5 206 13 240 9 202 208 245 174 0 3 76 2 396]

表中数据即为十进制的机器语言音乐程序当执行完该过程后音乐程序送入到了77O开始的内存单元。调用音乐程序的过程如下：

```
TO  MUSIC  : DATA
IF  : DATA=[ ]  STOP
. DEPOSOT  768  FIRST  : DATA
. DEPOSIT  769  FIRST (BF  : DATA)
. CALL  770  768
MUSIC  BF (BF  : DATA)
END
```

该过程第三句是将音阶送入768单元；第四句是将音长送入769单元，然后通过第五句调音乐子程序使嗽叭发音，第六句是递规调用MUSIC过程使嗽叭能连续不断的发出声音。

调用本过程的方法为：

MUSIC [225 160 228 160 205 160 192 160 171 160 152 160 140 160 128 160 114 160 114 160 102 160 95 160 84 160 75 160 68 160 62 160]

其中表中的数据是音乐数据，奇数位是音阶，偶数位是音长。音阶及音长以数据的关系如下：

音阶	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5
	225	288	205	192	171	152	140	128	114	102	95	84	75	68	62

音长：　1／4 拍　　1／2 拍　　1 拍　　2 拍　　4 拍

1／4 拍	1／2 拍	1 拍	2 拍	4 拍
30	70	110	160	255

270. LOGO 语言如何调用汇编语言

LOGO 语言调用汇编语言的目的是为了加快程序的运行速度。调用分为两步。第一步：用.DEPOSIT 命令把机器语言写入电脑的内存，一般可写在内存单元第三页（768）或从 $ 99A0（39328）开始的一些单元。.DEPOSIT 的命令格式为：

. DEPOSIT　<地址>　<算术表达式>

功能是把表达式的值送入地址单元。也可使用调 DOS 命令从磁盘中把汇编语言调到存贮器中。命令格式为：

DOS　[BLOAD　汇编文件名]

当程序已装入内存后就可以用.CALL 命令调用了。命令格式为：

. CALL　<程序启动地址>　<参数地址>

其中程序启动地址为：程序开始运行的首地址，参数地址为：程序需要数据的存放地址。

271. 如何用键盘快速画图

当我们要在屏幕上画一幅画时，要不断的输入 LOGO 命令，以控制图标按要求移动、尽管 LOGO 作图命令已很简练但仍然要输入很多命令不但费事而且容易出错误。为此向读者介绍一个简单的程序能使你作图的速度更快，更直观，更有情趣。

程序的主要思想是用键来控制画标移动。程序共用了六个键，其功能分别为：

U 键	抬笔	D 键	落笔
F 键	向前	B 键	向后
R 键	右转	L 键	左转

程序清单如下：

```
TO  OD
1: MAKE  "CH  RC
   IF  : CH="U  THEN  PU
   IF  : CH="D  THEN  PD
   IF  : CH="F  THEN  FD  1
   IF  : CH="L  THEN  LT  10
   IF  : CH="R  THEN  RT  10
   IF  : CH="B  THEN  BK  1
   GO  1
END
```

272. 如何打印 LOGO 绘出的图形

当读者用 LOGO 绘制出图形以后，不仅想从屏幕上看到它，也一定想通过打印机把图形打印出来。下来就介绍一下打印的方法。

在开机前，先将 DOS3.3 盘插入驱动器,然后开机，当 DOS 引导完后，再进入 LOGO，即键入：LG 进入 LOGO 运行方式。然后开始绘图，当绘图结束后，用图形存盘命令将图形存到磁盘上，命令如下：

?SAVEPICT ″MYPICT ；MYPICT 为图形文件名。或用 DOS 命令，如：

?DOS [BSAVE MYPICT ，A $ 2000，L $ 2000]上述两个命令的效果是一样的，只是文件名稍有区别。当用 SAVEPICT 命令时，LOGO 系统在存盘时自动在文件名后面加上一个后缀.PICT。使文件名最后变成：MYPICT.PICT。

完成图形存盘后退出 LOGO。简单的方法是送机后重新开机调入 DOS，或 CTRL−RESET、6、CTRL−P，用热启动调入 DOS。

当重新调入 DOS 后，用 BLOAD 命令把图形调入内存，并打开打印机。

] BLOAD MYPICT.PICT

然后键入：

] PR#1，起动打印机，接着键入 CTRL−Q，打印机就开始开印图形了。

273. 当图形画错了如何擦除

这里所说的画错了，并不是所有的都错了，只指刚画过的一笔。擦除也不是全擦除，只是擦除刚画错的那一笔。

在用图标作图时画错是难免的，而 LOGO 没有提供一条擦除的命令，但可以通过改变颜色的方法完成擦除。一般在进入 LOGO 后，LOGO 自动定义画笔为白色，背景为黑色。画图时，当某一笔画错后，立刻改变画笔颜色，使其变成黑色，再按相反的方向重复这一笔，画错的那笔就被擦去了。如：

? FD 100 此笔画错了
? PC 0 定义画笔为黑色
? BK 100 画错的那笔被擦去
? PC 1 定义画笔为白色
? LT 45 输入正确的命令

其中 PC 为改变画标颜色的命令，它后面的数值与颜色的关系如下：

0	1	2	3	4	5	6
黑	白	绿	紫	桔红	蓝	补色

274. 如何改变 LOGO 字符的显示方式

在中华学习机中 LOGO 语言没有改变字符显示方式的语句，使字符不能够反相显示或闪烁，这无疑是一个缺陷。为了弥补这一缺陷，使字符显示变的生动活泼，可以用下列过程加以实现。

过程 NORMAL 是将字符按正常方式显示；过程 INVERSE 是将字符以反相方式显示；过程 FLASH 是将字符以闪烁方式显示。使用时需要哪一种显示，就调用哪个过程就可实现预定效果。

```
TO  NORMAL
. DEPOSIT  232  255
END
TO  INVERSE
. DEPOSIT  232  0
END
TO  FLASH
. DEPOSIT  232  64
END
```

275. 如何把 LOGO 图象存入磁带

中华学习机的 LOGO 语言，可以把绘制完的图象存盘，但不能直接存带，这给没有磁盘的用户带来了不方便。为解决这一问题，谨向读者提供两个过程。过程 TSAVE 用来把图形存带，过程 TLOAD 用来把图形从磁带中取出装入计算机。

```
TO  TSAVE
. DEPOSIT  60  0
. DEPOSIT  61  32
. DEPOSIT  62  255
. DEPOSIT  63  63
. CALL  65229  0
END
```

```
TO  TLOAD
. DEPOSIT  60  0
. DEPOSIT  61  32
. DEPOSIT  62  255
. DEPOSIT  63  63
. CALL  65277  0
END
```

在执行 TSAVE 过程前应先把录音机连好，按下 PLAY 键和 REC 录音键。在执行 TLOAD 过程时还应注意录音机的音量和音频旋钮应调置适当的位置。

十二、中华学习机硬件部分

276. 中华学习机在电路结构上有何特点

CEC−I 型中华学习机是 APPLEⅡe 型微机的兼容机，与其它冯 · 诺依曼(Von Neumann) 型计算机一样，具有如图 276−1 的基本结构形式。

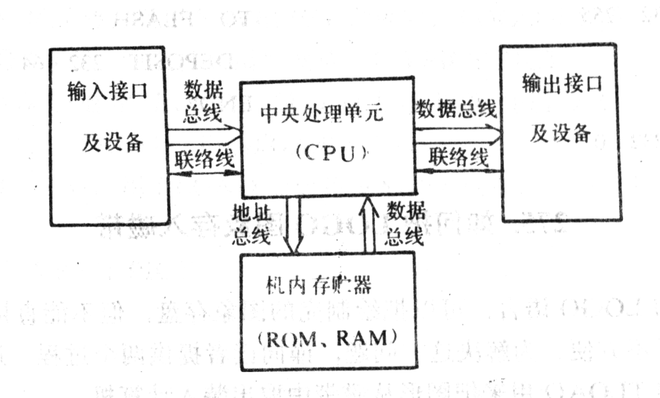


图 276−1 冯 · 诺依曼型计算机基本结构框图

但是，中华学习机和 APPLEⅡ微机有着与众不同有结构特点，主要表现在输入、输出接口电路上。APPLEⅡ微机主板上开设了八个接口插槽（亦称扩展插槽）；中华学习机虽然在主板上只开设了一个接口插槽，但可通过短路插塞的放置位置为该插槽设定不同的插槽号，用于与 APPLEⅡ微机各插槽对应。这些插槽可以插接各种用途的接口卡，用来构成适合各种应用环境的微机系统。这使得 APPLEⅡ微机和中华学习机具有很大的结构灵活性，成为该类微机的最突出优点之一。目前世界上流行的各种接口卡(用于 APPLE Ⅱ微机或中华学习机的) 已有上千种。

上述扩展插槽引线都是 50 线的。50 根引线连接了主机的绝大部分信号。其中包括：地址总线、数据总线、CPU 的各种控制线、时钟及四种电压电源等(如附图 7)。当需要利用 APPLEⅡ微机或中华学习机对一些设备进行控制或采集数据时，只要设计出一个相应的接口卡，插在插槽上用于与外界沟通信息，编排出控制程序就可实现实用的微机控制系统。

APPLE Ⅱ微机与中华学习机的另一个特点表现在它的显示方式上。虽然该类机 CPU(6502)的时钟仅为 1MHz，但是由于 CPU 是在时钟周期的一半时间内使用存储器，而另一半时间是显示电路使用存储器(如图 276−3 所示)，使得其工作速度不亚于其它 8 位微机(如 TRS80，CPU 为 Z80，时钟为 4MHz)。

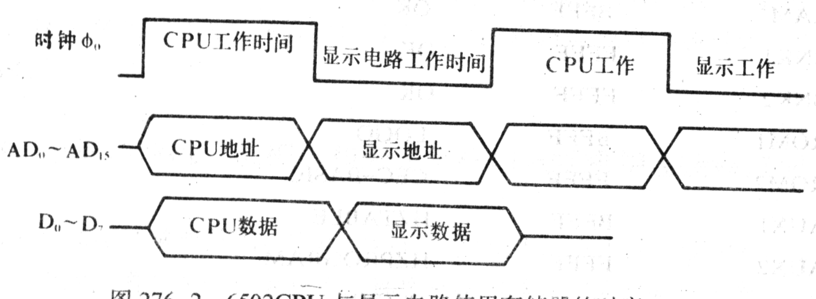


图 276-2　6502CPU 与显示电路使用存储器的时序

中华学习机是装有汉字系统和汉字库的 APPLEⅡ微机，它与 APPLEⅡ微机基本是全兼容的。

277. 为什么说中华学习机是 APPLEⅡe 微机的兼容机

APPLEⅡ微机在国内的主要机型有 APPLEⅡ型、APPLEⅡ+(plus)型和 APPLEⅡe 型。其中 APPLEⅡ型是早期产品，性能上有许多缺点。APPLEⅡ+型是在 APPLEⅡ型的基础上做了一些改良的机型。它们的硬件电路完全一样，仅在主板走线设计和选用器件上有些变化。APPLEⅡe 型机在硬件电路上有很大变化。它采用了高集成度电路，仅用 35 片集成电路就代替了 APPLEⅡ和 APPLEⅡ+型的 107 片集成电路。在功能和可靠性上大为提高。APPLEⅡe 型集成度的提高主要是使用了两片大规模集成电路：MMU 和 IOU，一片可编程阵列逻辑电路 PAL 以及 8 片 64k×1B 的动态 RAM (4164)。这些芯片的使用是 APPLEⅡe 型机的主要改良之处，并为其主要标志。CEC-I 型中华学习机共用集成电路芯片 45 片，其中有 27 片与 APPLEⅡe 型机的功能一样，其它 18 片包括 2 片 64K×4B 的动态 RAM(50464)和用于处理汉字、字库、控制磁盘驱动器以及彩色制式转换的芯片。中华学习机其实只是在 APPLEⅡe 型机的基础上增加了硬汉字电路及字库，并将磁盘驱动器的接口卡电路做到主板上。除此之外还实现了 NTSC 制对 PAL 制彩色制式的转换及射频信号调制。所以说中华学习机是 APPLEⅡe 型微机的兼容机。

278. 购置中华学习机应如何挑选

若希望购置一台称心如意的中华学习机，可在挑选时做以下几方面的检测。

1.主机自检：将中华学习机的射频输出头与一台彩色电视机天线插口用电缆连接好，打开彩色电视机电源开关和中华学习机电源开关。中华学习机在刚开机时发出“嘟”的一声响，说明扬声器及有关电路正常，主机无大故障。按下 CTRL-RESET-TEST 三键再放开（最后释放 TEST 键），主机将运行自检程序。屏幕上显示对主机 RAM 和 ROM 检测的结果。正常情况下应显示：

MEMORY-TEST

TIMES: 000

RAM	BFFF	OK
BNK1	FFFF	OK
BNK2	FFFF	OK
ROM1	BFFF	LOGO
ROM2	FFFF	CEC-BASIC
AUX1	BFFF	HZTABLE
AUX2	FFFF	HZPROGRAM
AUX3	5D99	CECWL

RAM 和 ROM 测试完毕后程序将自动进行彩色测试。这时，屏幕上应出现 16 条不同颜色的竖带。若无彩色或色彩不正都为不正常。无彩色时，可将主板上开关 SW1（见附图 9）拨向“OFF”一边。若仍无彩色则主机有故障。

2.驱动器接口检测：将磁盘驱动器 20 芯电缆插头插在主机驱动器接口插座上。驱动器内插入 DOS 3.3 磁盘。打开主机电源开关载入 DOS。用 DOS 命令“CATALOG”查看磁盘目录。再换一张空盘，用“INIT HELLO”命令初始化这张磁盘。上述动作均正确则驱动器接口电路完好。

3.键盘检测：依次按下键盘各键，屏幕显示应与键入字符一致。各键上下应自如，不能相互卡住。个别键有卡住现象时可将键壳摘下，用细砂纸轻轻打磨键壳四个突出边缘，然后装回。

4.扩展插槽检测：最好使用 Z80 卡进行测试，因为 Z80 卡几乎使用了扩展槽上的所有信号线。将 Z80 卡插在扩展插槽上；磁盘驱动器内放上 CP／M 操作系统磁盘。启动磁盘后若能正常进入 CP／M 操作系统则插槽各信号线正常。

5.电源测试：用各种 DOS 命令使磁盘驱动器工作起来，同时观察中华学习机面板上电源指示发光二极管的亮度。电源负载能力强时，发光二极管的亮度无变化、不闪动。否则亮度随驱动器动作而有明显变化。

上述各项测试完毕后，用手触摸电源附近的机壳部分应无明显温升。

这样挑选完的中华学习机一般是可靠的。

279. 中华学习机为什么要进行彩色制式的转换

中华学习机是 APPLE Ⅱe 型微机的兼容机。而 APPLEⅡ微机的设计者考虑到本国采用副载波频率为 3.5795MHz 的 NTSC 制式的彩色电视机和监视器，所以为 APPLEⅡ微机设计了频率为 14.318MHz 的晶体主振荡电路。将 14.318MHz 的信号 4 分频后可得 3.5795MHz 的彩色副载波信号；而将 14.318MHz 信号 14 分频后便得到频率约为 1MHz 的 6502CPU 所需的时钟信号。APPLEⅡ微机的显示信号产生电路部分将要产生副载波频率为 3.5795MHz 的彩色全电视信号。

我国彩色电视现采用的是副载波频率为 4.4336MHz 的 PAL 制式。原 APPLEⅡe 微机产生的视频信号不能适应 PAL 制彩电或监视器的工作条件。做为 APPLEⅡe 微机兼容

机的中华学习机为能够使家庭的彩色电视机用做显示器，就必须将主机产生的 NTSC 制视频信号转换成 PAL 制视频信号。否则，PAL 制彩色电视机或监视器将不能显示中华学习机发出的彩色图象。此外，中华学习机的射频信号调制电路(RFM)还要将 PAL 制的彩色全电视信号(视频信号)调制到射频信号上，以便彩色电视机可通过天线来接收中华学习机的图象信号。

280. 中华学习机、录相机和天线怎样与电视机相连接最方便

电视机的天线插座经常要分别与中华学习机、录相机、室内外天线相接，插头拔来插去很不方便。这里为读者介绍一种电视天线插座连接方法，可省去许多麻烦。而制作十分简单。

找一只高频的铁氧体磁环(内径 10 毫米外径 20 毫米)，按图 280−1 所示用 0.5～0.8 毫米线径的漆包线绕 4 个线圈，每个线圈的圈数都为 5～10 圈。4 个线圈分别与电视机天线插座、录相机射频输出插头、中华学习机射频输出插头、室内外天线插头相连接。连接好后就不用再拔下了。

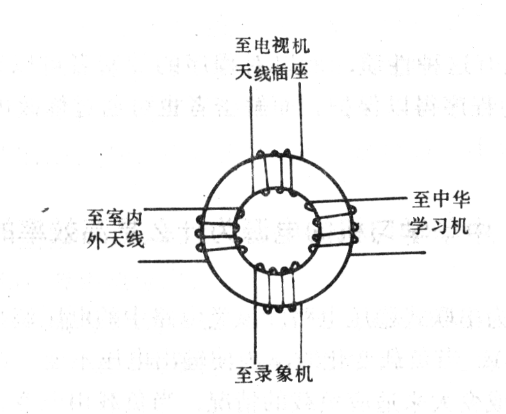


图 280−1 天线转接器电路图

整个电路若能装在一个小盒中并安装上相应的输出插头和输入插座就更好了。

281. 为什么说中华学习机是内存与外设统一编址型的计算机

计算机的 CPU 在工作过程中除了要与内存贮器之间交换数据，有时还要与外设之间交换数据。有些 CPU(如: Z80、8080 等)具有专门的输入输出指令，通过这些指令对接口电路进行控制，从而实现对外设数据的交换。另一些 CPU(如:6800、6502 等)没有专门的输入输出指令，它们是将通向外设的接口电路做为内存一样处理；这些接口的地址是与内存贮器地址统一编排的。使用 6502 CPU 的中华学习机，就属于这种计算机。

中华学习机 CPU (6502)的寻址范围是 0～65536，即 0～64K。用 16 进制数表示，则是从 $0000 到 $FFFF。其中从 $0000 到 $BFFF 一段地址为内存贮器 RAM 所占用；从 $D000 到 $FFFF 一段地址为监控 ROM 与 BASIC 解释程序等 ROM 所占用。而从 $COOO 到 $CFFF 一段 4K 范围的地址是留给外设接口电路使用的。这段地址通

常也被称为I／O地址。

282. 为什么按“RESET”键后RAM中的数据不会丢失

使用过TRS-80或H-01型等微机(CPU为Z80)的读者知道，当这类微机在工作中若操作者按了“RESET”键后，RAM中的数据和源程序就都丢失了，必须重新键入或载入才行。但是中华学习机不是这样。当你按过“CTRL-RESET”之后，状态从程序中退出而进入BASIC，RAM中的数据和源程序并不会丢失。这是因为在大存贮容量(16K～64K)的微机中都使用了动态RAM(如：4116、4164等)。这类RAM在工作时要不断地进行刷新才能将数据保留下来。TRS-80等微机的Z80CPU具有刷新信号，它控制着动态RAM不断刷新。而当复位信号RESET到来时Z80CPU将处于初始状态，且不产行刷新信号。这样，由于RAM得不到刷新信号而将数据丢失。中华学习机的CPU使用6502，这种CPU不产生刷新信号。主机板上的动态RAM是依靠显示地址发生器的信号进行刷新的。当用户按下“CTRL-RESET”键时，虽然6502 CPU也要初始化，但因显示地址发生电路并不因此而停止工作，这使得RAM仍在不断地刷新。RAM内的数据就被保留下来。

正是由于中华学习机有这种性质，所以对程序的加密者可以通过修改RAM内的“RESET向量”等使自己的程序得以保护。而解密者也可通过修改内存而对磁盘文件进行解密。

283. 中华学习机的电源为什么是高效率的

普通的直流稳压电源为串联式稳压电路。该类电路中的电压调整晶体管与负载形成串联形式（如图283-1所示）。当负载变化时，为使输出电压不变，调整管要改变自己的内阻使自己两端的电压变小或变大来适应负载的情况。当负载用电变化范围过大时，调整管两端就需要有足够大的电压余量。这势必使得很大一部分电能消耗到调整管上，变成热量。

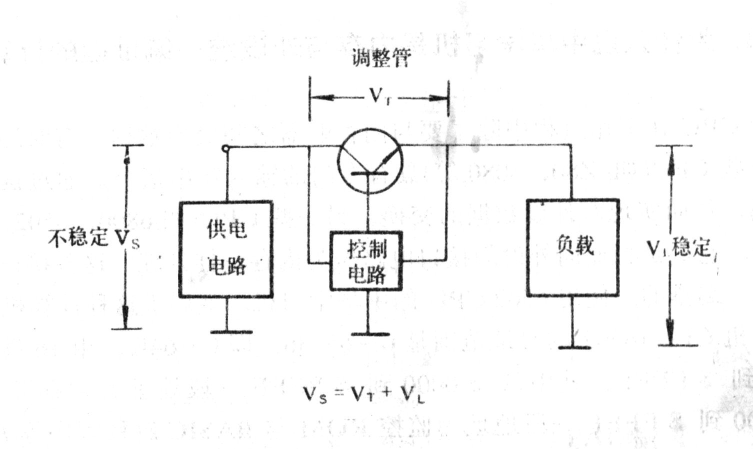


图283-1　串联式稳压电源稳压原理

中华学习机的电源为开关式稳压电源电路。它是利用晶体管的开关特性在较高的频率(大约 15KHz～30KHz)下向负载供电。当负载需要大电流时，开关管打开时间长些，反之打开时间短些。工作在开关状态下的晶体管，或两端电压近乎零，或内部电流等于零。所以其上的电能消耗极小。这样，电源的效率就提高了。

中华学习机开关电源由于采用的工作频率较 50Hz 的市电频率要高许多，所以变压器使用了体积很小的磁芯式开关变压器。同时，滤波电容的电容量也大为降低。整个电源体积很小，功率和效率却很高。其工作原理可见附图 8 。

其中 T1 为开关管，由桥式整流 BR1 和滤波电容 C1 向其提供约 310V 的直流电。控制电路使 T1 反复开关，通过 KT1 使 L2～L5 感应出相应的高频交流电。经 D1～D4 整流，C2～C5 滤波形成各种直流电压。晶体管 T2 采样输出电压，通过控制电路实现对 T1 导通占空比的调整来达到稳压的效果。

284. 中华学习机电源输出的四种电压各用在何处

中华学习机的开关式稳压电源输出四种电压，它们是 +5V， +12V， −5V， −12V。

其中 +5V 输出可向负载提供的最大电流为 2A。 +5V 用于中华学习机主机及部分不带电源的外设(如磁盘驱动器)中的 TTL 集成电路和其它电路。中华学习机主机板上有四十多个 TTL 电路，它们的工作电压都为 +5V。此外，打印机卡、Z80 卡等接口卡的电路也都是 TTL 电路，这些电路的供电电压也为 +5V。这就是为什么 +5V 电压供电电路要提供 2A 电流的原因。

−5V 电压输出最大电流可达 100mA。该电压主要是向中华学习机磁带数据放大电路的运算放大器 (MC1458) 提供负电源。

+12V 电压最大输出电流为 1A。 +12V 电压的最大耗电处是磁盘驱动器。磁盘驱动器的主轴电机、步进电机和磁头读、写放大电路都需要 +12V 电源。中华学习机主机的晶体振荡电路也使用了 +12V 电压。

−12V 电压最大输出电流为 100mA。在磁盘驱动器电路中，磁头读出放大电路 (MC3470) 芯片内部的差分放大器需要 ± 12V 双电源。

以上四种电源电压都被引到主机板的扩展接口插座上，以备外设使用。四种电源全部输出时，中华学习机电源的耗散功率为 25 瓦。

285. 中华学习机电源最易损坏的元器件是什么

中华学习机在正常使用当中最易损坏的元器件是电源输出端的整流二极管。这些整流二极管在损坏时的表现一般是呈短路状态。但有时是二极管的性能变差，高频情况下反向恢复速度变慢或反向截止不够。当整流二极管损坏时，电源表现为无输出电压或输出电压过低且不可调整。

中华学习机的主机电源是开关电源，其开关振荡频率为 30KHz 左右。由于开关变压器次级得到的波形基本上是连续的矩形脉冲，其基波频率在 30KHz 左右，而高频分量频率则几倍或几十倍于基波。这就要求整流二极管能在大电流(+5V 整流管达 2A 以上)高频

率下工作。工作条件相当苛刻，对二极管的要求较高。大电流要求二极管内 p–n 结面积大，而面积大会使 p–n 结的高频开关特性变差。所以在开关电源的输出端，+5V 整流二极管是由 2～3 只二极管并接使用的。当有一只二极管短路性损坏时，电源就会由于短路保护电路的作用出现无输出电压的故障现象。而当某整流二极管的高频开关特性变差，二极管在反向偏置时不能完全截止，造成有一部分交流分量输出。而这一交流分量又被滤波电容短路掉，所以输出的直流电压就会因这部分损耗而降低。既使调整输出电压调节电位器也不能将输出电压恢复。

所以，在遇到电源输出为零或过低时，主要检查的元器件应是整流二极管。

286. 中华学习机电源出故障时怎样应急检修

因为中华学习机电源最常见故障为正常工作时突然无输出或输出电压过低。所以仅就这两种故障现象叙述如何采取应急措施。

电源无输出电压的原因往往是电源输出整流二极管损坏。这些损坏一般是呈短路现象，所以在检查时可用万用表的 x1 欧姆档在电路板上直接逐个测量各整流二极管的正反向电阻，若发现某一二极管的正反向电阻都为零，则可断定该二极管损坏。但是对于+5V 的整流二极管，要用电烙铁烫下后再测量。这是因为+5V 电压输出电流大，电路上采取两只整流二极管并联使用。当其中一只二极管发生短路性损坏时，在线测量检查损坏二极管是无济于事的。

电源电压过低的原因往往是电源输出整流二极管的高频特性变坏所至。因电源的+12V、−5V、−12V 电压整流二极管都是单只的，所以当这些电压过低时只须将相应的二极管更换即可。而+5V 电压整流二极管并非一只，所以要逐个用电烙铁烫开，测量输出电压是否正常，以此判断损坏的二极管。

如果+5V 整流二极管中有一只短路性损坏，在手头无该型号二极管时可暂将损坏的二极管拆掉；这样便可排除短路故障。因为主机一般不会满负苛耗电，常规电流远小于 2A，有一只整流二极管便可正常工作。

如果是+12V 整流二极管损坏，可将损坏的二极管拆下，同时将+5V 整流二极管中烫下一只暂且代用，可应一时之急。

−12V 整流二极管最易造成人为性损坏。这往往是由于在插接磁盘驱动器电缆插头时，将插头接反所至。这时，−12V 整流二极管可用−5V 整流二极管代替。但磁带信息读入电路将不能正常工作。

287. 怎样正确使用“CILLIN Ⅱ”检测软件

APPLE – CILLIN Ⅱ 检测软件是专为检测 APPLE Ⅱ 微机而设计的。它的功能非常完全，可用来自动对 APPLE Ⅱ、APPLE Ⅱ+、APPLE Ⅱe 及中华学习机等微机的主机及外设进行全面检测。CILLIN Ⅱ 的基本使用环境如下：

1、主机至少要有 16K RAM，用于存放 CILLIN Ⅱ 检测程序和系统使用的零页内存、堆栈、键盘缓冲区、第三页向量及文本显示缓冲区。

2、应至少接有一台磁盘驱动器，用来载入磁盘上的 CILLIN Ⅱ检测程序。

3、若需要打印出检测结果，应在第 1 号槽上接有打印机。

4、对于 PC－5500 之类分离式键盘的 APPLE Ⅱ兼容机，还应将键盘电缆接好。

当 CILLIN II 主程序被载入后将自动运行。它首先显示一个菜单，提示共有六类基本测试功能供用户选择：

1、检测所有 RAM(包括扩展的)。

2、检测所有 ROM(包括接口卡的)。

3、检测磁盘驱动器。

4、检测其它项目。包括键盘、显示器、CPU、打印机等等。

5、自动地连续检测 1 至 4 项内容。

6、设定是否打印检测结果。

按 1 至 6 数字键则分别选中对应的检测功能，该功能模块将从磁盘上调入内存并自动运行。同时主菜单还提示用户，若按"X"键则退出此检测程序而进入 BASIC 状态。

在任何一项检测中，只要按"ESC"键便可回到主菜单。并且每做一步都是有提示的。使用起来非常方便，所以用法无须多谈。这里仅就检测当中显示的内容说一下它的意义，并以此来判断微机故障部位。

在检测 RAM 的过程中，若只在屏幕上半部显示所测 RAM 的地址，而下半部屏幕无显示则说明该段 RAM 无故障。当检测到 RAM 有故障时，下半屏幕的最左边一列显示有故障 RAM 的地址；第二列是该地址内原来的内容；第三列是检测时送进的新内容；第四列是读得的新内容；第五列是要送回的原内容；第六列是再读出的原内容。这时显示的第三、四列内容一般是不同的。

CILLIN Ⅱ在检测主板与接口卡上的 ROM 时，采取的办法是以 2K 为单位把各单元的内容进行异或(EOR)。如果运算结果是它所认可的值，则屏幕显示该 ROM 内存贮的内容名称。如："APPLESOFT"(表示是 APPLESOFT BASIC 解释程序)，"AUTOSTART"(表示是自启动监控程序)等。否则显示"UNKNOWN"(表示不认得)。但是，由于各种类型 APPLE Ⅱ微机及兼容机 ROM 中的内容不尽相同，所以当 CILLIN Ⅱ计算出 ROM 内的异或值时应将其记录下来，以便以后检测故障时对照。检测 ROM 时的显示结果如以下样式：

$ F000	28	UNKNOWN
(开始地址)	(异或值)	(ROM 内容)

CILLIN II 在检测磁盘驱动器时需要一张空磁盘，然后按它所需格式进行格式化，再做各项读、写检测。做该项检测时一定不要忘记将放有 CILLIN II 软件的磁盘从驱动器内拿出来。

288. CILLIN Ⅱ检测软件不能载入时怎么办

CILLIN Ⅱ检测软件可用来检测中华学习机各部分工作情况。如果某部分有故障，可借助此检测软件测试到。但是，若中华学习机的某些部分发生故障，使得 CILLIN Ⅱ程序不能正常载入运行该如何处理呢？下面向读者介绍一些常规的排除故障简法，以便迁

到这种情况时参考操作。

CILLIN Ⅱ软件不能载入的原因可分为几个方面。其中只要有一个方面故障，则载入过程就不能完成:

1、主机故障，CPU 没有工作。

2、主机 RAM 故障。

3、驱动器接口电路故障，驱动器没有工作。

4、驱动器故障，不能正常读磁盘。

5、磁盘数据被破坏，不能正常载入。

这些故障的现象并不是完全一样的，通过对故障现象的分析可以排除或发现故障点，为对微机的进一步检测奠定基础。

若为主板 RAM 中有损坏的，更换 RAM 即可。若为磁盘数据破坏，可再重新考贝一张磁盘。若为主机电路、驱动器卡及驱动器故障，可参阅本书中有关题目进行检修。

289. 怎样定义“全角”汉字输入功能

中华学习机在汉字状态下被定义了三种输入方式，即：“ASCII 码方式”、“拼音方式”、“区位方式”。它们分别对应于键盘上部的 F1、F2、F3 这三个功能键。中华学习机出厂时 F4（键码＄14）和 F5（键码＄06）两个功能键未被定义。本文向读者介绍一种将 F4 键定义为“全角方式”功能键的方法。在“全角”输入方式下，用户可通过直接按键盘上的英文字符和阿拉伯数字来输入汉字方式下的这些字符和数字，而不必去查区位码表。这可使输入工作加快许多。

在监控状态下将表 289－1 所列机器码键入从＄8000 开始的内存中，然后输入“8000G”命令，再退出监控到汉字状态。此后，每当按 F4 键时便可进入“全角”输入方式。

表 289－1 “全角”功能机器码

8000－	A9	15	8D	91	03	A9	80	8D	8040－	03	A9	02	8D	B6	03	A9	FF
8008－	92	03	A9	2B	8D	8F	03	A9	8048－	4C	AB	C3	60	1F	1F	1F	1F
8010－	80	8D	90	03	60	A9	23	85	8050－	1F	1F	1F	1F	1F	1F	1F	1F
8018－	FB	A9	80	85	FC	20	6E	C3	8058－	1F	1F	1F	1F	1F	1F	1F	1F
8020－	4C	AB	C3	7F	B4	EC	7F	CD	8060－	1F	1F	1F	1F	1F	1F	40	41
8028－	C3	3A	20	AD	C9	03	C9	C1	8068－	42	43	44	45	46	47	48	49
8030－	90	19	E9	C1	AA	BD	4C	80	8070－	4A	4B	4C	4D	4E	4F	50	51
8038－	8D	BE	03	BD	66	80	8D	BF	8078－	52	53	54	55	56	57	58	59

290. 检修中华学习机时使用逻辑笔或示波器哪个更方便

使用逻辑笔检修中华学习机及外设电路，有时要比使用示波器更为方便。原因在于:

1、中华学习机电路的信号除几个时钟信号具有固定的周期外，其它所有各点的信号都是非固定周期的。若用示波器观察各点波形，示波器因不能同步而使波形无法稳定下来，观察效果不能让人满意。充其量只能说明有否波形存在，这就与逻辑笔具有同样功能

了。

2、电路中往往存在只有一个单脉冲的情况。迂此情况时，示波器不能记录下该脉冲。而逻辑笔有记忆功能，能将是否产生脉冲的情况反映出来。

3、示波器的显示器与信号探头远离，使用时又要注意被测点，又要观察显示器，十分不方便。而逻辑笔小巧，且探头与显示装在一起。

此外，由于电路的损坏往往不是逻辑上出现错误，而是彻底损坏，使信号不再是脉冲而是“固 1”或“固 0”。这就使得详细反映波形内容变得无意义。

但是，当检修中华学习机主机及外设的模拟电路部分(如：电源电路、视频信号输出电路、磁带输入放大电路、磁盘驱动器的磁头读写放大电路等)时，示波器就要比逻辑笔用起来更好。因为这些部位电路的信号不仅需要看波形的有无，还要测量波形的幅度。

由于逻辑笔有前述的一些优点，而且价格低廉，所以往往是检修微机及外设必备的工具。

291. 怎样自制逻辑笔

在检修中华学习机及其外设时，逻辑笔可说是必不可少的工具。对于未有逻辑笔或买不起逻辑笔的微机用户，如果能自制一支逻辑笔将是一件乐事。

制作逻辑笔使用的元器件都是很普通的，在一般电子商店就可买到。其电原理图见图 291-1。

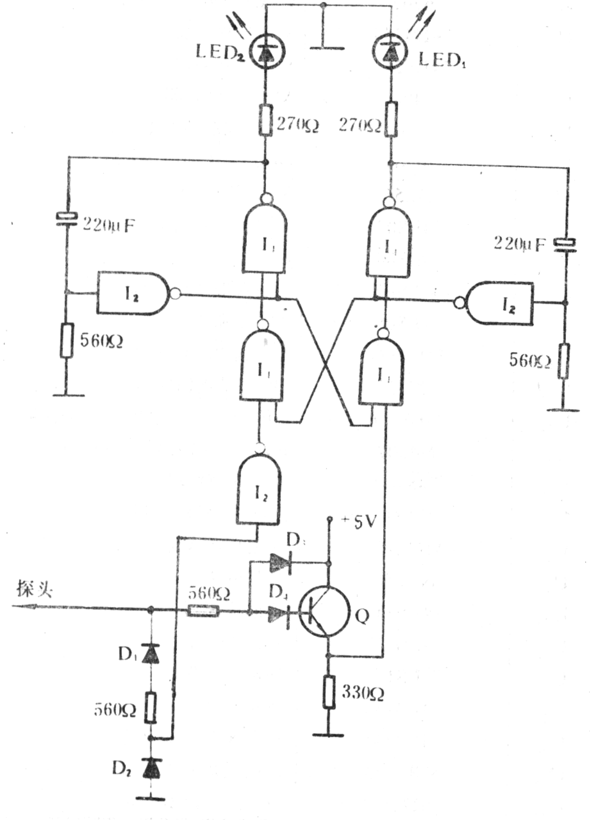


图 291-1　自制逻辑笔的电路图

图中 I1 和 I2 为两片 74LS00。LED1 和 LED2 为两只发光二极管，其工作电流越小越好。D1～D4 为 2CK 型二极管，Q 为 3DG 型三极管。两只电解电容为 220uF／6V 或耐压 6V 以上规格的。所有的电阻都为 1／8W 的。整个电路安装在一个宽 2 厘米，长 15 厘米的印刷电路板上，然后装入一个长方形或圆柱形的塑料外壳中。引出一个探头，一根 +5V 电源线和一根地线。使用时，地线要与被测电路地线相连。+5V 电源线可接被测电路的 +5V，也可另接。用探头碰触被测点。若被测点为“1”(高电平)时，LED1 亮而 LED2 灭。被测点为“0”(低电平)时，LED2 亮而 LED1 灭。被测点若为脉冲信号，则两只发光二极交替闪亮。若被测点为高阻态或探头开路，两只发光二极管都灭。

该电路的缺点是不具备脉冲记忆功能。当被测点存在单个脉冲时 LED1 和 LED2 将交替闪动一下，然后就保持恒“1”或恒“0”的显示。

292. 中华学习机主板上的元器件是怎样编号的

为了使中华学习机主机的安装、调试、分析和维修方便；中华学习机的设计者与生产厂家将主机板上的全部元器件编了号，并印在主板上(如附图 9 所示)。现将各元器件编号、型号及名称列于下面表 292–1 中。

表 292–1　元器件对照表

元器件类别	编号	元器件型号	元器件名称
集成电路	U_1	6502	CPU(中央处理单元)
	U_2	74LS244	8 总线驱动器
	U_3	74LS244	8 总线驱动器
	U_4	N4006	MMU(存储器管理单元)
	U_5	50464	64K×4B 动态 RAM
	U_6	50464	64K×4B 动态 RAM
	U_7	27256	32K 字节 EPROM
	U_8	74LS00	2 输入端 4 与非门
	U_9	74LS04	6 反相器
	U_{10}	74LS02	2 输入端 4 或非门
	U_{11}	74LS08	2 输入端 4 与门
	U_{12}	74LS374	三态输出 8D 触发器
	U_{13}	2732	4K 字节 EPROM
	U_{14}	74LS166	8 位串／并入串出移位寄存器
	U_{15}	N4007	IOU(输入输出管理单元)
	U_{16}	74S109	双 J–K 触发器
	U_{17}	N4008	PAL(可编程阵列逻辑)
	U_{18}	MC1458	双运算放大器
	U_{19}	NE558	时基电路
	U_{20}	74LS251	数据选择器
	U_{21}	74LS138	3–8 译码器
	U_{22}	74LS138	3–8 译码器
	U_{23}	74LS138	3–8 译码器
	U_{24}	74LS245	双向 8 总线驱动器
	U_{25}	KB3600	扫描式矩阵编码器
	U_{26}	2716	2K 字节 EPROM

续表 292-1

	U_{27}	74LS164	8 位串入并出移位寄存器
	U_{28}	74LS175	4D 触发器
	U_{29}	74LS74	双 D 型触发器
	U_{30}	TCA650	色信号调制器
	U_{31}	74LS139	双 2-4 译码器
	U_{32}	74LS174	6D 触发器
	U_{33}	CECWL Ⅰ	汉字库 ROM
	U_{34}	CECWL Ⅱ	汉字库 ROM
	U_{35}	27256	32K 字节 EPROM
	U_{36}	74LS02	2 输入端 4 或非门
	U_{37}	74LS125	4 三态缓冲器
	U_{38}	74LS10	3 输入端 3 与非门
	U_{39}	74LS323	8 位双向通用移位寄存器
	U_{40}		PROM(256 字节)
	U_{41}	74LS259	8 位可编地址锁存器
	U_{42}	74LS174	6D 触发器
	U_{43}	74LS132	2 输入端 4 与非施密特电路
	U_{44}	74LS05	6 OC 反相器
	U_{45}	NE555	时基电路
射频调制器		RFM	射频调制器
晶体三极管	Q_1	2N4258	PNP 型三极管
	Q_2	2N4258	PNP 型三极管
	Q_3	MPSA13	达林顿管
	Q_4	2N3904	NPN 型三极管
	Q_5	2N3904	NPN 型三极管
	Q_6	2N3906	PNP 型三极管
	Q_7	2N3904	NPN 型三极管
	Q_8	2N3906	PNP 型三极管
	Q_9	2N3904	NPN 型三极管
	Q_{10}	2N3906	PNP 型三极管
石英晶体	Y_1	$14.318MH_Z$	
	Y_2	$4.43MH_Z$	
开关	SW_1		$3.58MH_Z$ 吸收开关
插座	J_1		50 线扩展槽插座
	J_2		9 芯游戏口插座
	J_3		5 芯录音机口插座
	J_4		20 线磁盘驱动器插座
	J_5		视频信号监视器电缆插座
	J_6		射频信号电视天线插座
	J_7		主机电源插座
	J_8		扬声器插座
	J_9		20 线键盘插座
	J_{10}		槽号设定跨接插座

293. 中华学习机主板上元器件怎样按功能分区

中华学习机主板上的元器件按其功能不同可大致分为五个区域(如附图 9 所示)。

区域Ⅰ:基本系统工作区(主控区)。该区域对应于附图 1 和附图 2 的电原理图。其中 U1(6502 CPU)是整机的核心。U4(MMU)管理着整机的存储器系统。U7(27256 EPROM)中固化有系统监控程序、BASIC 解释程序、磁盘启动程序和部分汉字管理程序。U15(IOU)主要用于控制屏幕显示、管理显示缓冲区,同时还产生录音机和扬声器输出信号。U17(PAL)产生系统各种定时信号和控制信号。U5 和 U6(50464)构成 64K × 8B 的动态 RAM,为整机的随机存储器。

区域Ⅱ:接口电路控制工作区。该区域对应于附图 3 的电原理图。其中 U21、U22、U23(74LS138)用于产生各个接口(扩展槽口、键盘接口、扬声器接口、游戏接口、录音机接口等)的控制与选通信号($C000—$C7FF)。U19(NE558)和 U20(74LS251)用于连接游戏杆,接收游戏杆的信号。U25(KB3600)用于扫描键盘产生键值地址码;而 U26(2716EPROM)可将键值地址码转换成键值 ASCII 码。U24(74LS245)则用于向扩展槽 J1 的数据总线上输送和接收数据信号。

区域Ⅲ:视频信号产生电路及制式转换电路区。该区域对应于附图 4 的电原理图。其中 U27(74LS164)用于产生视频信号中的亮度的色度信号。U28(74LS175)、U29(74LS74)和 U30(TCA650)则用于 PAL 制式的转换。

区域Ⅳ:汉字系统工作区。该区域对应于附图 5 的电原理图。其中 U33、U34 为固化有汉字点阵的 ROM。U35(27256EPROM)中固化有汉字系统管理程序。

区域Ⅴ:磁盘驱动器接口电路区。该区域对应于附图 6 的电原理图。其中各电路功能在有关题目中叙述。

了解了中华学习机主板上的分区,对于检修机器、查找故障点是大有好处的。

294. 怎样使用逻辑笔快速准确找出故障点

逻辑笔在检测数字电路信号时,可显示出信号是高电平(+2.4V～+5.0V)、低电平(0V～+0.8V)或连续脉冲。有些逻辑笔还可以记忆单个脉冲。

中华学习机主板上的集成电路基本上都是 TTL 电路。在正常工作时,这些集成电路各管脚上的信号绝大多数是连续脉冲。有些管脚上则分别是高("1")或低("0")电平。中华学习机故障往往是由于主板上的集成电路损坏而造成的。这些集成电路损坏时的表现常见于:原为脉冲信号的管脚变成了固"1"或固"0"信号。这就为使用逻辑笔检测故障点提供了可能性。

附图 10 示出了中华学习机主板上各集成电路管脚的正常信号状态。其中"H"表示高电平逻辑"1","L"表示低电平逻辑"0","P"表示连续脉冲,"X"表示该管脚为空(即未接信号)。

在寻找故障点之前,应先从开机时和开机后显示器及扬声器的反应大致估计出故障所在部位或区域;然后根据附图 10 中所标注集成电路各管脚的信号情况用逻辑笔逐一检

测。发现有不正常的管脚时，要按本书后附的电原理图查找与该管脚相连的所有集成电路芯片，将这些芯片及相关电路芯片更换一下，便可把损坏的电路找出，故障即被排除。

295. 高分辨率图形显示不正常的原因有哪些

高分辨率图形显示异常的故障现象一般有三种形式。第一种是在 BASIC 状态下键入“ HGR ”或“ HGR2 ”命令后根本不能进入高分辨图形方式，屏幕上只是出现一些竖条和一些低分辨率图形色块。这类故障往往是 U15(IOU)损坏造成其“SEGB”信号不正常或 U13(2732 EPROM)损坏造成的。第二种故障现象是键入“ HGR ”或“ HGR2 ”命令后可以进入高分辨率图形方式，但是不能清屏。屏幕上有许多很规律的竖直方向亮道子。这类故障往往是 U17(PAL)损坏所至。第三种故障现象是进入高分辨率图形方式后，在屏幕上出现一些无规则的亮点或亮块。这类故障是由于主板上的动态 RAM 有个别单元或个别位损坏后呈恒″1″状态在屏幕上形成亮点。该故障有时通过自检可检测出故障所在，但有时自检也查不出来。

296. 如何从屏幕和喇叭的表现迅速估计主机故障大致部位

当中华学习机主机发生故障时，通过刚开机时屏幕及喇叭的反应是能够马上估计出故障发生的大致部位的。这是由于屏幕显示与系统的时基电路、显示电路、CPU 工作电路有着直接的联系；喇叭的发声则与系统的监控程序运行情况有关，同时也与 CPU 接收键盘字符有关。

一般常见故障现象有如下几种:

1. 开机后屏幕无光、喇叭无声，电源指示发光二极管不亮。
2. 开机时屏幕只闪烁一下，然后无光。喇叭不响。
3. 开机后屏幕全亮；喇叭不响。
4. 屏幕显示杂乱字符或低分辨率图形；喇叭不响。
5. 喇叭发出连继的“嘟嘟”声；屏幕乱。
6. 喇叭只有“嘎”的一声响。
7. 喇叭有“嘟”的一声；屏幕无亮。
8. 屏幕有些不稳的斜条纹；喇叭不响。
9. 屏幕显示不稳定；喇叭有“嘟”的一声。

这些故障现象中，故障 2、3、4、5、8 都是主机系统发生的“致命性”故障，即故障点是在时基或 CPU 及地址、数据线上。

现象 1：故障在开关电源部位，或主板上有电源短路点。

现象 2：故障发生在时基电路，此时主板上无 14MHz 的脉冲。

现象 3：故障发生在 CPU 本身或 CPU 的地址、数据、读写控制线。

现象 4：故障发生在 RAM 地址多路开关电路(MMU 即 N4006)。

现象 5：故障发生在 RAM 电路及其地址译码电路。主板 I／O 电路故障时也有此现象。

现象 6：此为 RAM 区发生故障。

现象 7：故障发生在视频输出电路、视频输出电缆或监视器。

现象 8：CPU 的控制信号线有开路或短路现象。

现象 9：监视器行、场不同步。

通过以上故障点部位的初步判断，再进行进一步的仔细检修要比无目的检修更加方便。所以，对故障点的初判是十分重要的。而初判又是可以通过屏幕和喇叭的表现迅速得到的。

297. 开机时屏幕只闪烁一下后无其它反应的故障原因是什么

总的来说，开机时只是屏幕闪烁一下，喇叭无声，其故障原因一般是由于主机的时基电路没有工作。至使 CPU 无时钟脉冲，显示地址发生器也没有工作。

先检查晶振电路。用万用表直流电压档测一下+5V 和+12V 是否正常；如果电源电压正常，则可用逻辑笔看 Q2 集电极有无脉冲存在。

1、如果 Q2 集电极有脉冲，则检查 U37–11(即芯片 74LS125 的第 11 脚)有无脉冲。若无脉冲且 U37–13 为低，则 U37 损坏。若有脉冲，则应检查 U16–6、U16–7、U16–10 各脚是否有脉冲，如果无脉冲则有可能 U16 芯片损坏了。

2、如果 Q2 集电极无脉冲，则应检查 Q1 和 Q2 的工作状态是否正常。在正常情况下 Q1 和 Q2 的基极电位为+5V，发射极电位在+5.5V 左右。如果这些直流电位与正常值相比差别过大，则是 Q1 或 Q2 损坏。否则是石英晶体(14.318MHz)失效或接触不良所至。

中华学习机各时钟产生电路由一片大规模可编程逻辑门阵列 PAL(或 PLA)和一片 74S109(双 JK 触发器)组成。该电路最易出现的故障是 PAL(N4008)损坏，同时该故障也是 APPLE IIe 微机最常见故障。

由于没有 14MHz 的脉冲，所以一系列的时钟就都未产生，CPU 和显示地址发生器都不能正常工作。但是在开机的瞬间，显示地址发生器和视频信号产生电路由于加电可能会产生一个脉冲，这一脉冲被送到监视器电路使屏幕闪烁一下。

298. 开机后即进入监控是何原因

中华学习机在开机后，如果正常，则进入 BASIC 状态；屏幕上出现提示符“]”。但是当主机电路有某些故障时，开机后则进入监控状态；屏幕上出现提示符“＊”。该故障在两种情况下存在。一种情况是无磁盘驱动器时，开机后直接进入监控。另一种情况是有磁盘驱动器，开机后载入 DOS 时进入监控。

第一种情况往往是因为低位 RAM 中或 BASIC 解释程序被固化的 ROM 中有某个集成电路损坏造成的。这是由于监控程序与 BAISC 解释程序都使用了零页(做为 CPU 的一种寻址方式)、第一页(做为系统堆栈)、第三页(设置暖启动矢量和 BASIC 入口矢量) RAM。读一下监控程序(从入口地址 ＄FA62 开始)会发现，在程序进入 BASIC 之前，监控仅向低位 RAM 置数。若不进入 BASIC，则反复读键盘呈等待状态。但是，程序进入 BASIC 之后，要频繁使用(读、写)低位 RAM。若 RAM 损坏则可能使 CPU 中的程序计

数器 PC 内容发生改变，程序运行跑出 ROM 区而进入 RAM 区。RAM 区内大部分内容为“ 00 ”，这一机器代码为 CPU 的“BRK”指令，即软中断(或强迫中断)。该中断的中断矢量存放于地址为 $FFFE 和 $FFFF 的 ROM 单元，其内容分别为“40”和“FA”。于是 CPU 从 ROM 监控程序区的 $FA40 单元取指令，执行监控程序。此段程序则是显示 CPU 中各寄存器内容，然后返回监控初始化程序段，等待键盘键入字符。如果是固化 BASIC 解释程序的 ROM 损坏，程序执行的正常过程也会被破坏，而进入 RAM 区取得“BRK”指令。但是如果是高位的 ROM 损坏，机器虽然会进入 BASIC 状态，出现提示符“]”。不过，当从键盘输入回车符时，机器还会进入监控状态。

第二种情况则往往是由于 RAM 损坏或不稳定造成的。在载入 DOS(DOS3.3 版)时，DOS 程序是分三级逐步载入高位 RAM($9600～$BFFF)区的。并且以后 DOS 要常驻高位 RAM 区。如果高位 RAM 损坏，DOS 程序必被破坏，则出现与 BASIC ROM 损坏时同样的情况——进入监控。如果是低位 RAM 不稳定，或有些监控未使用但载入 DOS 时却使用了的低位 RAM 单元损坏，则会引起 DOS 载入的地址或内容不正确。这样，当机器将控制权交给 DOS 后，同样会进入监控状态。

对于这类故障，只要将有故障的 RAM 或 ROM 查出更换即可。

299. 为什么有的中华学习机的主板上还增加一块小电路板

有些中华学习机打开上盖后会发现在 U5 和 U6(50464 RAM)的插座上插有一个小电路板，该板上焊有 8 个 4164 RAM(64K×1 位)芯片。这是由于原中华学习机使用的 50464 RAM 是 64k×4 位的两片动态 RAM 芯片。该芯片由于受到进口的限制不能向厂家供货,有些中华学习机的生产厂家就使用 8 片 64K×1 位的 4164 (或 4864 等)来代替两片 50464。由于 50464 是 18 脚芯片，而 4164 是 16 脚芯片，所以要将 8 片 4164 做在一块电路板上，将主板上所需各信号引到该电路板，并以插接的形式将该电路板插到 50464 芯片的管座上。这样代换后中华学习机功能不受影响，电源耗电也未增加多少。只是当机器受较强振动时易发生电路接触方面的故障。

300. 中华学习机最易损坏的器件是什么

APPLE Ⅱe 微机及其兼容机 CEC-Ⅰ型中华学习机都是 APPLE Ⅱ微机最大改良型微机，它们使用的集成电路数量大为减少，集成度有很大提高。但是改良后的这类微机其时基电路则由一片可编程逻辑门阵列 N4008 (PAL) 芯片来承担。该芯片产生系统几乎所有的定时信号和时钟信号，还包括一些控制信号 (如：行、列选通信号等)。这无疑对 PAL 芯片的负载能力是一个考验。如果该芯片质量较差，则很快会损坏。另外，由于中华学习机的扩展接口插槽上 3.5MHz 和 7MHz 信号线与-5V 或-12V 靠近，而 3.5MHz 和 7MHz 信号又接至 N4008 的两个输入端。当用户在带电插拔接口卡时，-5V 或-12V 线就会轻易碰到 3.5MHz 和 7MHz 信号线，而使 N4008 永久性损坏。所以一般地讲中学习机最易损坏的器件就是 N4008。

N4008 损坏后微机的表现主要有下列几种：

1、开机后喇叭不响，屏幕无光。只在开机的瞬间屏幕闪烁一下。

2、开机后喇叭响，但屏幕不显示字符，只是一些杂乱的竖道子。

3、刚开机时一切正常，使用一段时间后屏幕左右两侧出现杂乱字符，屏幕中间显示的字符也有缺点的地方并不停地闪动。

迂到 N4008 芯片损坏时，只好更换一片。但在国内此片不易买到。实在搞不到时，可使用“GAL”（Generic Array Logic）通用阵列逻辑芯片。但该芯片价格昂贵，且需要将 N4008 输入输出逻辑关系确定后用 GAL 编程器写入该芯片。

301. 按键与显示字符不符的原因是什么

当屏幕上显示的字符与所按键不相符合时，造成这种故障可有两方面的原因。

第一、键盘接口电路传送给主机系统工作部分的键值码是错误的。

第二、主机系统工作部分传送给显示部分的显示码是错误的。

对于原因一，由于送入主机系统的信息就是错误的，所以系统不会做为命令接受。即在键入 DOS 命令或 BASIC 命令时，系统一定会做出“？SYNTAX ERROR”的反应。这往往是键盘扫描编码电路 U25(KB3600)故障或键码转换电路 U26(2716 EPROM)故障所至。

对于原因二，由于仅是送到显示电路部分的代码错误，所以不影响系统的正常工作和程序的运行。这种故障往往是显示字符发生路 U13(2732 EPROM)或 U14(74LS166)损坏所至。其中 U14 的损坏多见。并且损坏后的表现往往是工作频率跟不上，而不是彻底不能使用。这样，在屏幕上显示的字符或不对，或失去原来的模样。

302. 按一下键后出现一串相同字符是什么原因

一些中华学习机键盘容易出现“连发”的故障。即当用户按一下某个键时，屏幕上会出现一个以上的一串相同字符（应与键盘的重复功能相区别）。产生这类故障的原因一般分为两种。其一是新计算机，各键之间有时容易相互卡住或被上盖卡住。由于键盘的重复功能而出现“连发”现象。这种故障易被发现和排除。其二是中华学习机在使用了较长时间后，由于按键本身都是机械接触的，触点有积炭或杂质，键被按下时会产生反弹的机械杂波。为了克服这些杂波，键盘扫描编码电路在扫描到有键按下时要延长 8 毫秒再接收键值。但若按键触点接触不良现象严重，就会使编码电路误认为是一次以上按该键，于是产生这类故障。此时，可用四氯化碳或汽油清洗故障的触点加以克服。

当然，如果键盘扫描电路 U25(KB3600)损坏，也会产生该故障现象。

303. 监视器显示字符不清楚的原因是什么

监视器显示字符或图形不清楚的原因可能是中华学习机视频输出电路故障，也可能是监视器或视频信号电缆有问题造成的。

中华学习机视频输出电路如附图 4 所示。当晶体管 Q8、Q9 性能变差或损坏时，监

视器屏幕上图象暗淡、很不清楚，而且图象极不稳定。如果使用电视机显示，则若 W3 电位器调整不当或失效，也要影响显示效果。

如果是监视器方面的故障，则往往是监视器对比度、亮度和聚焦调整不好造成的。当使用电视机时，由于普通电视机荧光屏分辨率较低，或高频调谐器频率调整过偏，也会使图象显示不理想。

最不容易引起人们注意的故障点是视频传输电缆。主机与监视器间视频信号的传输全靠这条电缆。它应该是一条 75Ω 同轴电缆，其高频特性很好。若用普通金属屏蔽线代替同轴电缆，就会使视频信号的高频分量损失过多，造成显示图象不清晰，字符中间部位亮而左右两边缘处暗淡，好象有一个渐变的过程。

除此之外，视频产生电路的 U14(74LS166)不良，也会造成显示字符不清楚。

304. 彩色显示不正常的原因有哪些

使用彩色电视机或彩色监视器而无彩色，首先应检查中华学习机主板磁盘驱动器 20 芯电缆插座(J4)左边的开关(SW1) 是否置于″OFF″ 的位置。若 SW1 置于″ON″的位置，则色信号调制电路 U30 (TCA650)将停止工作，当然主机不会有彩色信号输出。

若显示的彩色不正常，则有可能是下面几处发生故障：

第一、色差信号编码电路 U28 (74LS175) 损坏，会使显示的颜色失真。

第二、电位器 W1、W2 调整不当，也会使颜色失真。

第三、副载波振荡电路工作频率偏离 4.43MHz 太多，彩色会不正常甚至失去彩色。这可通过调整微调电容 C41 来校正。

另外，如果电位器 W3 调整不当或晶体管 Q8、Q9 不良，会使图象信号太弱，造成彩色失真或无色，而且图象不稳定。

305. 中华学习机磁盘驱动器的结构是怎样的

我们知道，中华学习机的磁盘驱动器是一种单磁头的 5.25 英寸软磁盘驱动器。

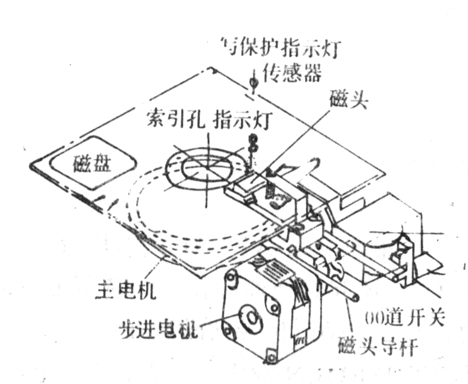


图 305−1　磁盘与磁头的相对位置图

微型个人计算机的外存贮器多采用磁存储方式，有磁带、软磁盘、硬磁盘。其中磁带存储是在一条长带上读写数据，是一维存储方式。软磁盘存储是在一个园平面上读写数据，是二维存储方式。硬磁盘则相当于在一个园柱体上读写数据，属于三维存储方式。所以从存储容量与读写速度上排队的话，最大的应为硬磁盘，其次为软磁盘，最差的为磁带。

既然软磁盘存储方式是在一个园平面上读写数据，这就要求磁盘与读写磁头间有两个方向的相对运动。它们分别是：磁盘相对于磁头的轴向转动和磁头沿磁盘的径向移动。这样，磁盘驱动器内就存在两个传动机构，一是驱动磁盘

转动的部分，二是驱动磁头平动的部分。(如图 305−1 所示。)

软磁盘驱动器机械与电气的基本零部件有以下各部分:

1.主轴电机。带动磁盘转动。工作电压直流 12V。转速为 300 转／分，允许误差 ±1%。

2.步进电机。带动磁头移动。为四相步进电机。工作电压 12V。步距角为 1.8°。最大转角为 145°。

3.读、写、抹磁头。将数据读写到磁盘上。

4.主轴电机伺服电路。对主轴电机进行调速和稳速控制(电路见图 305−2)。

5.步进电机伺服电路。对步进电机进行控制。(电路见附图 11)

6.磁头读写控制与放大电路。控制磁头的读、写动作，并将磁头读出的数据或将要写入的信号放大。(见附图 11)

7.其它部分。包括写保护判断电路、索引孔判断电路、零磁道定位电路、磁头导轨、磁盘夹等。

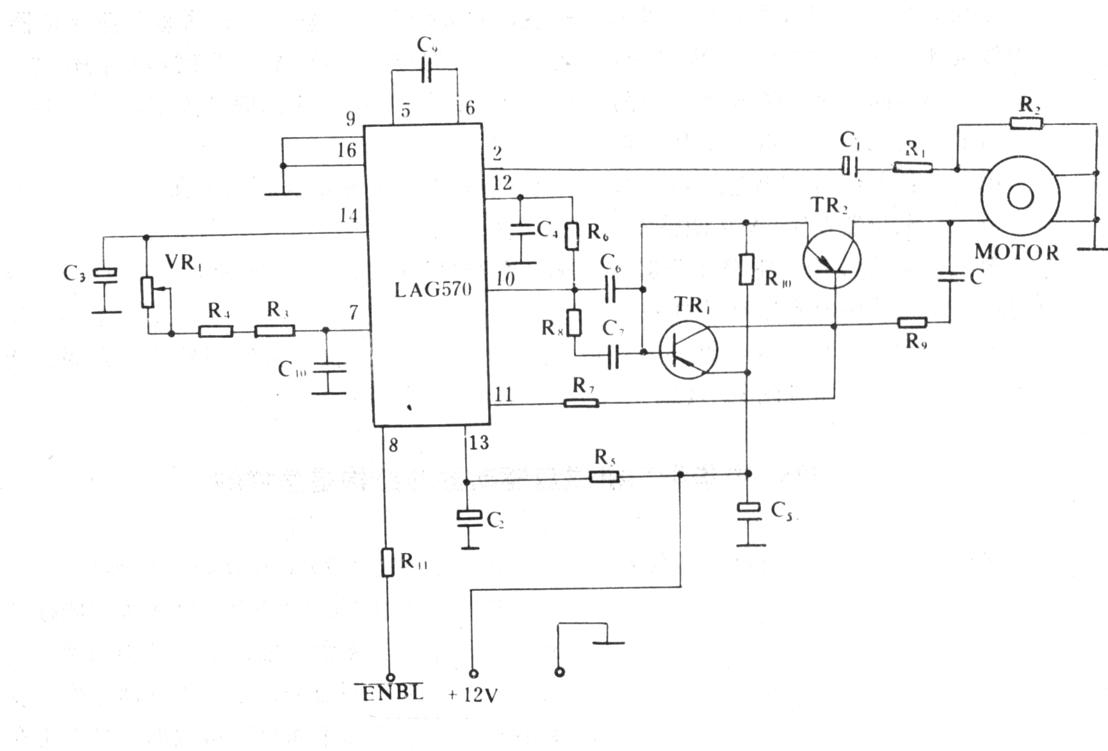


图 305−2 主轴电机控制电路

磁盘驱动器工作过程简述如下:

1.当驱动器被选中时，ENBL＝0(参见图 305−2)，集成电路 A1(LAG570)开始工作。晶体管 TR2 导通，向主轴电机提供工作电流。同时 A1 还起着稳速的作用。调节电位器 VR1 可改变 A1 的工作状态，从而调整主轴电机的转速。主轴电机带动磁盘旋转。

2.步进电机控制信号 $\theta0 \sim \theta3$ 通过三态 OC 反相器 D4(ULN2003; 参见附图 11)成为驱动步进电机步进的四相信号 $\varphi A \sim \varphi D$。步进电机带动磁头移动，将磁头移至预定的位置。

3.读操作时，磁头与磁盘相对运动产生的磁感应信号由磁头输入到放大电路B1(MC3470)中。再从B1第10脚输出，经三态门B4(74LS125)成为RD DATA串行信号送至磁盘驱动器接口电路上进行“串变并”处理。

4.写操作时，WR REQ＝0，使B4的第5、6脚为“通”。WR DATA串行数据信号被送至磁头驱动电路A3(CA3146)的第2脚，经放大后进入读写磁头，将数据写到磁盘上。

306. 中华学习机磁盘记录方式是怎样的

中华学习机软磁盘的记录方式是使用NRZ的GCR方式。其中NRZ意为“不归零”制；GCR意为分组编码记录方式。

在磁盘驱动器中，数据信号要通过磁头写到磁盘上或从磁盘上读取信号。磁盘上涂有磁性材料；在磁头中电流的作用下，磁盘上的磁性材料要产生相应的永久磁场。当需要读回数据时，磁盘上的磁场在磁头中感应出电流，便建立了数据信号。但是，由于磁场是在磁头中电流发生变化时才能建立的；所以，磁盘的记录方式是当数据信号为“1”时建立磁场，数据信号为“0”时不建立磁场。另外，由于磁盘驱动器中只有一个磁头，不可能将主机的8位数据同时写到磁盘上，而必须把并行的8位数据变成串行的数据，一位一位地写到磁盘上。

图306-1示出了串行数据对磁盘的写过程与读过程中各信号变化的情况。

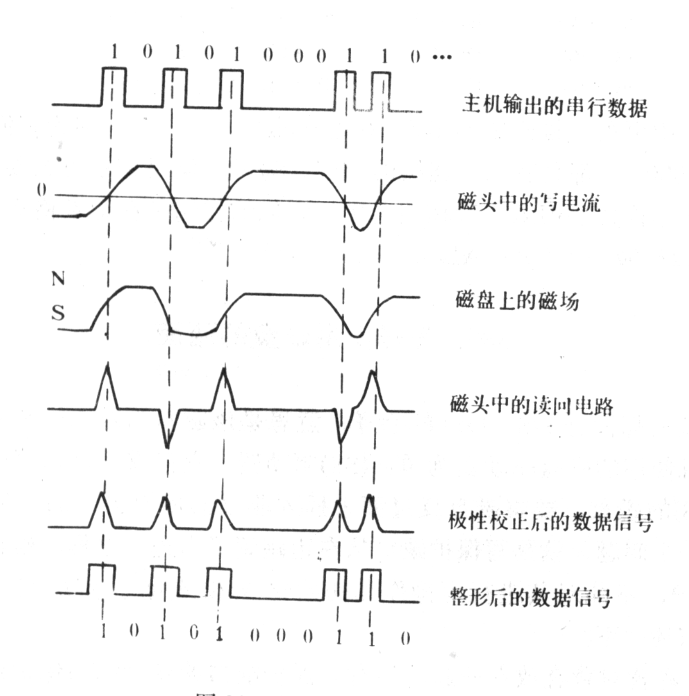


图306-1 NRZ记录过程

这种记录方式的缺点是：若有大于2个连续的“0”位连在一起时，数据读出电路将不容易分辨出“0”的个数。所以，一般不采用直接的NRZ方式。中华学习机则是将数据重新进行编码，使数据中不出现大于2个的连续的“0”。这就是所谓GCR方式。这种记录

方式要求:

1.一个字节的最高位必须是“1”。

2.最多允许连续一对“0”，并且在一个字节中只能出现一对。

3.至少要有一对连续的“1”。

中华学习机的DOS是采取6/2编码的GCR方式。即：将每个字节8位先分成高6位和低2位的两部分，三个字节的低2位部分再组成一个新的6位部分。然后，将这些6位部分编成8位码（编码详见表306–1）。这些8位码都是满足记录方式要求的。这样处理后，一个256字节的数据块就变成342个字节。

表306–1　GCR 6/2编码表

00→96	10→B4	20→D6	30→ED
01→97	11→B5	21→D7	31→EE
02→9A	12→B6	22→D9	32→EF
03→9B	13→B7	23→DA	33→F2
04→9D	14→B9	24→DB	34→F3
05→9E	15→BA	25→DC	35→F4
06→9F	16→BB	26→DD	36→F5
07→A6	17→BC	27→DE	37→F6
08→A7	18→BD	28→DF	38→F7
09→AB	19→BE	29→E5	39→F9
0A→AC	1A→BF	2A→E6	3A→FA
0B→AD	1B→CB	2B→E7	3B→FB
0C→AE	1C→CD	2C→E9	3C→FC
0D→AF	1D→CE	2D→EA	3D→FD
0E→B2	1E→CF	2E→EB	3E→FE
0F→B3	1F→D3	2F→EC	3F→FF

从表306–1中可见，编码中不出现16进制的“1”和“8”，因为它们的二进制数分别为“0001”和“1000”，都具有三个连续的“0”。另外，满足记录方式要求的两个8位码“D5”和“AA”作为保留字而未出现在编码中，它们的作用是在记录格式中区别“地址域”(D5 AA 96)和“数据域”(D5 AA AD)。

307. 怎样使用磁盘的两面

一般在磁盘上都标有DS/DD的字样，意思是该磁盘为双面(DS)双密度(DD)磁盘。但是中华学习机使用的磁盘驱动器为单面(SS)驱动器，它仅有一个读写磁头。所以对于具有双面标志(DS)的磁盘，要将磁盘反过面来插入驱动器使用另一面。但是将磁盘反过来使用时会迂到一个间题，就是写保护缺口从左边跑到了右边。这样，磁盘驱动器就会认为该磁盘有写保护，不能对其进行写操作和初始化(即格式化)。解决这一间题的办法有几个，下面分别叙述一下：

1.将两块磁盘背对背叠放在一起，沿有写保护缺口的磁盘的写保护口边，用剪刀将无缺口的磁盘剪下一个缺口来。两边都如此操作后，这两张磁盘就都是两边有写保护口的双面使用的磁盘了。

2.磁盘的另一边不用剪写保护口。将驱动器外壳打开，在磁盘插入时的写保护口处可

找到驱动器上固定的一个发光二极管和一光敏三极管。用万用表的Ω档检查它们各自两端的正反向电阻。正反向电阻差别较大的那个便是发光二极管。在光敏三极管(只用了两个极)两个极的连线处焊出两根引线，在需要写无缺口磁盘的背面时将两根引线短接到一起即可。

3.无须将磁盘的另一边剪出缺口。打开驱动器外壳，找到光敏三极管位置。在做磁盘写操作时(最好是在考贝备份资料磁盘时)，用一手电筒照射光敏三极管。写完磁盘后将驱动器复原。

4.方法同3。只是不用手电筒，而是用一只金属螺丝刀将光敏三极管两引线焊点短路。

上述方法中的2.3.4法考贝好的磁盘具有只能读不能写的特点。对于一些长期保留的程序库、数据库、汉字库等磁盘，因磁盘已满且仅读不写，这样处理是最合适的。这既节省了磁盘，又保护了数据，而且对不了解实情的人还有一定的保密作用。

308. 怎样给磁盘驱动器增加"强写"与"唯读"的功能

在上一问中我们说明了，对没有剪缺口的磁盘背面如何正常读写。但那毕竟是应付一时。这里我们介绍一种改装磁盘驱动器电路的方法，可长期使用，并具有"强写"、"正常"、"唯读"三种功能。

磁盘驱动器的写保护控制电路如图308-1所示。

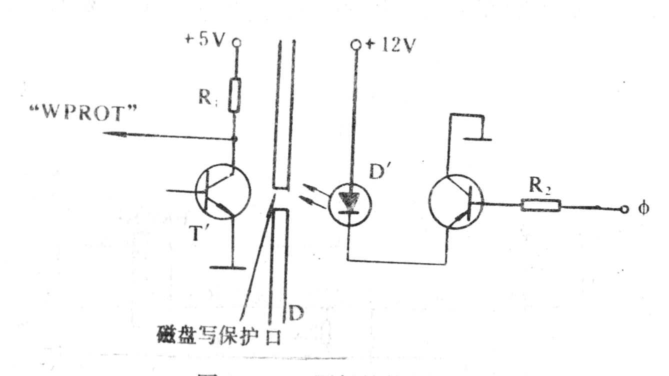


图308-1 写保护控制电路

当发光二极管D′的光线照到光电三极管T′上时，T′导通而使写保护信号WPROT=0。反之，WPROT=1。驱动器卡及驱动器内电路就是由"WPROT"信号来控制着写允许和写数据电路。如果我们能人为地控制这一信号的高低，就可达到我们改装的目的。

改装时，找一只1×3的开关，安在驱动器适当的位置，然后按图308-2接线。

当开关置于①位置时，"WPROT"恒为+5V，为"1"状态。此时，磁盘驱动器只能读，不能写，谓之"唯读"状态。

当开关置于②位置时，"WPROT"恒为"0"，磁盘驱动器将不管磁盘上有无写保护和写保护口，都能强行将数据写到磁盘上。我们称它为"强写"状态。

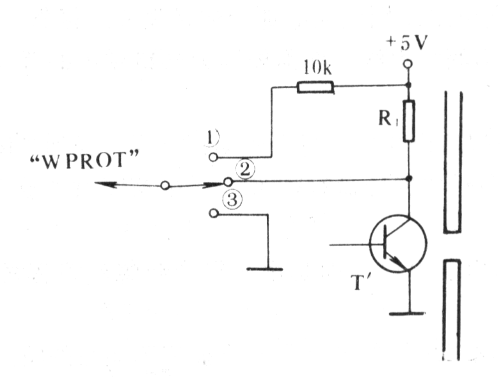


图 308-2　写保护控制改装电路

当开关置于③位置时，为"正常"状态。磁盘能否写上数据取决于是否贴有写保护签。

这样改动过的磁盘机会给您带来许多方便。同时也带来一定的危险性。所以，在平时应将开关常置于"唯读"或"正常"状态，以防误操作造成损失。

309. 怎样给磁盘驱动器再增加一层保护

为了防止因操作者的疏忽造成对磁盘信息无法弥补的破坏，往往在磁盘上贴一封写保护签。这种方法对正常的磁盘驱动器确实可起到磁盘写保护作用。但是，如果驱动器发生故障，则有可能将贴有写保护的磁盘加以破坏，造成所谓"毁盘"现象。

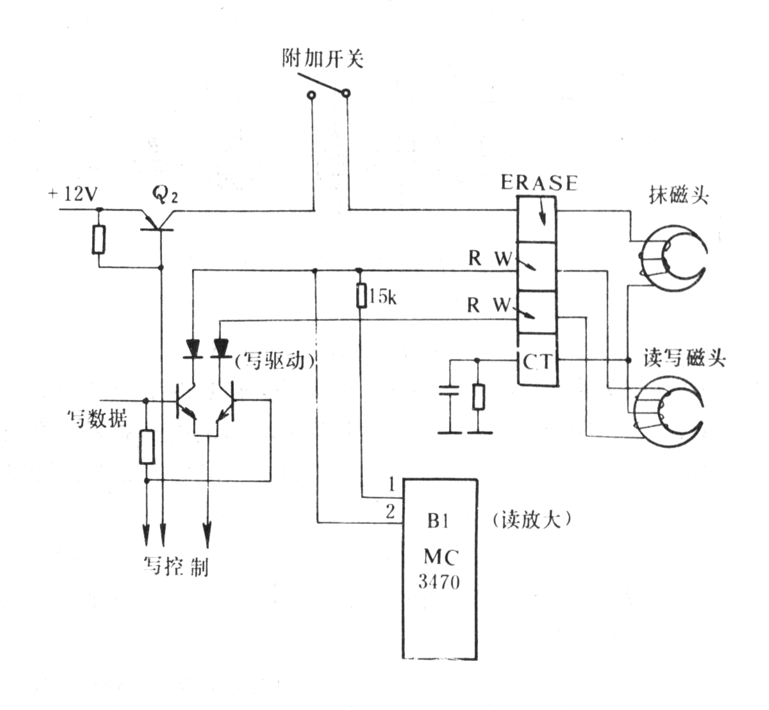


图 309-1　磁头写电路的改造

这里介绍一种方法，可在磁盘机发生故障时也不会造成"毁盘"。

从附图 11 中磁盘驱动器写磁头与抹磁头的电路分析可知；当读写磁头进行写动作时，必须由晶体管 Q2 提供大约 11V 的工作电压。当此工作电压不存在时，既使驱动器的写磁头控制与数据电路已工作，但仍不会有写数据电流流过磁头。如图 309−1 所示，Q2 是磁头写动作产生的关键，所以我们可以在 Q2 的集电极回路中串接一只开关。并将开关装在驱动器外壳上。当只进行磁盘读操作时，可将附加开关断开。只有在需要进行写磁盘操作时，再将附加开关闭合。这样处理后，虽然对操作者来说造成了一些麻烦；但是却可以确保在驱动器出故障时不会“毁盘”。一旦当磁盘驱动器出现不能正常读盘现象时，就不要再将附加开关闭合而去做写磁盘的操作。

310. 怎样给中华学习机装两只磁盘驱动器

中华学习机的磁盘驱动器接口电路是被做在主板上的。但是它的电路与 APPLE II 微机磁盘驱动器接口卡电路相比没有什么太大区别。所以从原则上讲，接两个磁盘驱动器是可行的。只是在接线上要下些功夫。

中华学习机的磁盘驱动器电缆与 APPLE II 微机的一样，共有 20 根导线。它们的分配如下：

地线(GND)：第 1、3、5、7 脚
+12V 电源：第 13、15、17、19 脚
+5V 电源　第 11、12 脚
−12V 电源：第 9 脚
步进电机驱动：第 2、4、6、8 脚
写请求(WRREQ)：第 10 脚
写保护(WPROT)：第 20 脚
读数据(RDDATA)：第 16 脚
写数据(WRDATA)：第 18 脚
使能端(ENBL)：第 14 脚

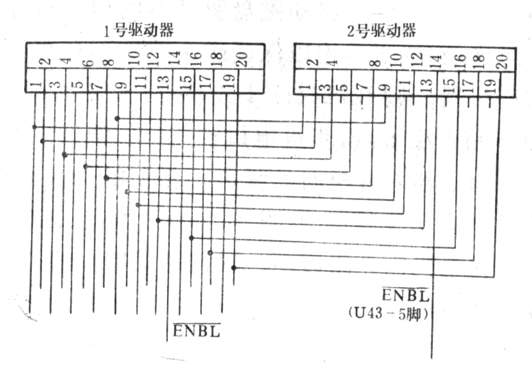


图 310−1　双磁盘驱动器接线图

这些接线中，使能端 ENBL 是用来控制磁盘驱动器工作的。当该信号为低电平时，所接驱动器的电路将被接通，各部分电路被启动。所以，当接两个磁盘驱动器时，除了 ENBL 接线不同外，其它 19 根连线是并接在一起的。附图 6 示出了中华学习机磁盘驱动器接口电路。

加接磁盘驱动器可按图 310−1 接线。此时，只需在主电路板上的接口电路处，从 74LS132 (U43，二输入端四施密特与非门)的第 5 和 11 脚引出一根线，接至 2 号磁盘驱动器 20 芯电缆插接头的第 14 脚。其余 12 根线与 1 号驱动线对应信号线相接。

311. 为什么开机时有的磁盘驱动器噪声大，有的噪声很小

首先我们说一下开机时磁盘驱器产生噪声的原因是什么。

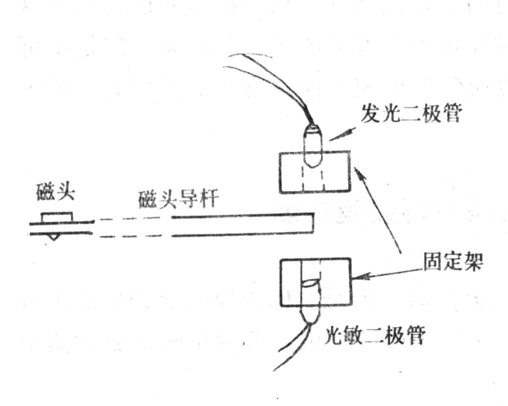


图 311-1　0 道定位装置

磁盘驱动器磁头是在步进电机的驱动下，沿磁头导轨在磁盘的径向上前后移动。而磁盘的第 0 磁道在最外圈。中华学习机开机时要求磁盘驱动器磁头定位到第 0 磁道载入 DOS。为了能使磁头可靠地定位到第 0 磁道，在磁盘驱动器结构上和 DOS 启动程序中都采取了一些相应的措施。在步进电机的中轴旁，有两块固定销子，它们使步进电机的活动范围在 145°之内。而其中之一销子使得磁头移位不至超出磁盘第 0 磁道以外的部分。开机时的噪声就是步进电机轴的突出部位与定位销撞击发出的。另一方面，在主机的 PROM 中固化有启动磁盘驱动器的启动程序。该程序在开机时要先让磁头向第 0 号磁道方向移动 80 步，用以保证磁头定位不至在第 0 磁道以内的部分。若开机前磁头停在第 10 磁道位置，则开机时磁头先移 20 步便回到第 0 磁道位置，而后面还要再移动 60 步。这 60 步将产生 60 次步进电机轴与定位销发出的撞击噪声。

为了消除开机时的噪声，新型的磁盘驱动器(低噪型)在磁头导杆的后端与固定架上安装了 0 道定位装置(见图 311-1)。当磁头欲移至第 0 磁道以外时，磁头导杆后端将固定架上的光电机构光路截断，磁盘驱动器上的控制电路使步进电机断电而不能继续转动，撞击噪声便可消除。

312. 中华学习机磁盘驱动器磁头最多可移多少步

磁盘驱动器的磁头是在步进电机的驱动下一步一步移位的。磁头移动的最大距离受到步进电机转角的限制。中华学习机磁盘驱动器的步进电机其最大转角为 145°；而通常对步进电机采用四相单四拍加电方式，则其步距角为 1.8°。这样，步进电机带动磁头可移动的步数为:

145 / 1.8 = 80.56 (步)

即通常所说的磁头移动 80 步。

中华学习机磁盘驱动器属于单面单密度驱动器。中华学习机 DOS 3.3 版本规定磁头每移动两步（即步进电机转两个步距角）为一个磁道。所以一张磁盘在格式化时最多可建立 80 / 2 = 40 个磁道，但通常只使用前 35 个磁道。

如果我们改变步进电机的加电方式，采用四相双四拍加电方式，则其步距角可减少至 0.9°。这样磁头移动的步数可达 160 步。步进电机以 1.8°的步距角移动，可实现所谓“1 / 2 轨”的读写；而以 0.9°的步距角移动，便可实现“1 / 4 轨”的读写。若再改变步进电机加电方式，可使其步距角进一步细分。理论上采用“细分技术”可将步进电机的步距角

减小到 1.8°／256。但实际上，由于中华学习机磁盘驱动器为单密度磁头，当磁道为 1／2 轨间距时，磁头已无法分辨两磁道的数据，细分的意义也就不大了。

313. 能否格式化一张 80 个磁道的磁盘

由于中华学习机磁盘驱动器磁头移动次数为 80 步，所以仅从这点上讲，格式化一张 80 个磁道的磁盘是有可能的。但是，中华学习机磁盘驱动器是单密度的，磁头的精度不够高，分辨力较低。当两磁道间距为正常磁道的二分之一时，磁头将不能正确分辨两相邻磁道(新出厂的磁盘驱动器其磁头偶尔有时可将两个 1／2 轨间距的磁道分辨开)。既使在所谓“1／2 轨加密法”中，也不过是利用正常磁道之间的部分建立磁道，而两磁道间的距离仍与正常磁道间距相同——即两个步距角。

由上述可见，若想增加磁道数，唯有提高磁头的精度。这里，我们介绍一种可行的办法。找一只 IBM PC 微机磁盘驱动器(双面双密度驱动器)的磁头，用其将 APPLE II 微机磁盘驱动器磁头换下。然后将 DOS 3.3 载入，进入监控状态将 $B7EE 单元的内容 01 改为 00。再回到 BASIC 状态用“ INIT HELLO ”命令初始化磁盘。如果成功，则说明此磁头可用。

下一步，便应修改 DOS。因为 DOS 3.3 的 VTOC (第 $11 磁道的第 0 号磁区)的 T／S 表(BIT MAP)中只有 40 个磁道的使用管理资料，即 DOS 3.3 最多只能管理 40 个磁道。所以比较好的方案是将一张磁盘的 80 个磁道分成前 40 道和后 40 道分别管理，而这两部分可分别代表一号与二号驱动器。这样，一个驱动器就可以当两个用，使费用降低一半。

314. 中华学习机磁盘的正常转速应为多少

中华学习机磁盘记录是采用 NRZ 方式，同时还要将每两位数据间形成 4uS 的间隔。这样每记录一个字节(8 位)就需要用 4uSX8＝32uS 的时间。

中华学习机的 DOS 规定磁盘记录格式如下：

地址域

10 个字节的同步头	D5 AA 96	8 个字节的地址	DE AA	EB

数据域

5 个 FF	D5 AA AD	342 个字节的数据	1 字节校验	DE AA EB

这是每个磁区(SECTOR)的记录格式。则每个磁区共有字节数为

(10+3+8+2+1)+(5+3+342+1+3)＝378

磁盘上每一磁道有十六个磁区(DOS3.3 版)，而第 0 号磁区前的同步头为 100 多个字节。我们又知道每记录一个字节要用 32uS 的时间，所以磁盘上每记录一个磁道，亦即磁盘每转一圈所用时间应这样计算：

32uS×(378×16+100)＝196736uS

＝196.736mS

则磁盘的正常转速为大约 5 转／秒＝300 转／分钟。

315. 磁盘驱动器应怎样进行调速

磁盘驱动器调速是指对驱动器主轴电机转速控制电路进行调整，从而实现将磁盘转速调整到正确值。这一工作，一般应借助驱动器调速软件来完成。

打开驱动器处壳，找到主电机控制板。一般驱动器内有两块印刷电路板。一块为模拟板，它上面有磁头读写数据与控制电路和步进电机驱动电路。另一块为主轴电机控制板，上面是主轴电机控制电路。该板往往反装在支架上面，而看不到元件面。但在该板上可找到一个转速调整孔，孔内有调速电位器 VR1(参见图 305−2)调节柄或调节孔。将有测速软件的磁盘放入驱动器，打开主机运行测速程序。用螺丝刀慢慢调节 VR1 并注意屏幕上显示的转速误差。当误差值调至“0”时便完成了调速工作。

图 315−1　驱动器测速卡图

有时如果手头无驱动器测速软件，或所具备的测速软件不适合该驱动器而无法使用(详见测速软件的使用一题目）时，对有些驱动器可采取这样的粗调方法：

在驱动器主轴电机的后面轴上固定有一惯性圆金属盘。盘上贴有一张画着两圈黑道子的图纸。(如图 315−1 所示)称为测速卡。两圈道子分别标有 50、60 两种数字。在日光灯下让驱动器转起来，看标有“50”的那一圈道子是否形成无闪动的一圈(因为市电频率为 50Hz)。连成一圈无闪动时则说明磁盘转速准确。否则说明转速偏离准确值太远。如果观察到闪动呈顺时针方向移，说明磁盘转速偏低；闪动呈逆时针方向移，说明磁盘转速偏高。在市电为 60Hz 的地区,应观察标有“60”的那一圈。

316. 使用驱动器测速软件要注意哪些问题

磁盘驱动器测速软件从测速方式上看可分为两大类。一类是利用磁盘上的索引孔和磁盘驱动器上的磁区定位机构对磁盘的转速进行测定。如：磁盘测速程序“SPEED TEST”。另一类则是通过磁头将标志写到磁盘上，再从读该标志次数的过程中计算出磁盘的转速。如：COPY 软件“LOCK SMITH4.1”中的测速程序“HIRES SPEED TEST”和 APPLE Ⅱ微机检测软件“CILLIN Ⅱ”中的“DRIVE SPEED TEST”。

在使用第一类测速软件时，驱动器应具有索引孔测定机构。而使用第二类测速软件时，一般都需要一张按要求格式化好的空盘，否则将把磁盘上原来的信息破坏掉。所以，可根据自已驱动器的情况（是否有索引孔测试机构）和磁盘的情况（手头是否有空盘）来决定使用哪类测速软件。

317. 能否通过增加磁盘转速来提高主机读写磁盘的速度

要想搞清楚这个问题，首先应了解驱动器卡及 DOS 中读写磁盘程序是如何工作的。下面我们以写磁盘为例来说明这一过程。

磁盘驱动器卡电路示于附图 6 中。

电路中的 74LS323 是一个 8 位通用移位寄存器。数据可并行或串行输入该寄存器，也可从该寄存器中并行或串行输出。串行移位可左、右两个方向进行。将要写入磁盘的数据先被并行打入 74LS323 中，在 CP 脉冲 (2MHz 的 Q3 脉冲)作用下串行地从 QA′端输出。在电路 74LS174 (六 D 触发器)和 P6A(可编程 ROM) 的共同作用下形成 NRZ 记录方式的一串信号。这一信号从 74LS174 的第二脚输出，做为“ WR DATA ”信号送至磁盘驱器的磁头写放大电路。整个过程是在 Q3 脉冲的作用下进行的。 Q3 脉冲是中华学习机时基电路产生的 2MHz 周期脉冲。 74LS323 中的 8 位数据，每形成一位 NRZ 编码要用 8 个 Q3 脉冲，即要用 4uS 的时间。这一时间已由电路决定，是决不会变化的。

为了与上述过程密切配合，DOS 的写磁盘程序也经过严格的设计。请看这一程序段：

```
          ORG      $B82A
             :
             :
      LDA   $C08D, X ;      测写保护。X 中为驱动器所在槽号×16。
      LDA   $C08E, X ;
      BMI   $B8B4    ;      写保护则转走。
      LDA   $BC00    ;      取第一个数据。
      STA   $26      ;      暂存数据
      LDA   #$FF     ;      准备写同步头 ( 5个#$FF )
      STA   $C08F, X ;      将 #$FF并行打入 74LS323
      ORA   $C08C, X ;      选通 74LS323 ，开始写盘
      PHA            ;      延时 3T
      PLA            ;      延时 4T
      NOP            ;      延时 2T
      LDY   #$04     ;      延时 2T
B84A - PHA           ;      延时 3T
      PLA            ;      延时 4T
      JSR   $B8B9    ;      写后四个 #$FF 。用 6T
      DEY            ;      用 2T
      BNE   $B84A    ;      用 3T
             :       ;      此后写 "D5 AA AD"
          ORG      $B8B9
      PHA            ;      延时 3T
      PLA            ;      延时 4T
```

```
STA    $C08D, X ;     将下一#$FF打入74LS323。用5T
ORA    $C08C, X ;     启动74LS323。用4T
RTS             ;     返回。用6T
```

从$B84A开始计算所用机器周期数共为40个T。而对于6502为1MHZ的时钟来说，1个T就是1uS，所以每写一个#$FF(8位一个字节)要用40uS。

以上分析说明，向磁盘读写数据的速度是固定的。通过加快磁盘转速非但不能加速读写，还会造成读写错误。

318. 驱动器调速时错调了模拟板上的电位器应如何恢复

有些用户在调整磁盘驱动器转速时，不仔细分析电路，将驱动器外壳打开，见到一个电位器就调。结果错把模拟板电路上的电位器(参见附图11)R28当作驱动器主轴电机调速电位器(见图305-2)VR1。调整后，磁盘转速没有改变，而驱动器却发生了读数据错误、不能引导磁盘、写数据错误并伴有"毁盘"等故障。

打开驱动器后，一般不易发现调速电位器VR1。因为它和其它元器件一起是背装到固定架上，我们只能看到印刷电路板的背面这些元器件的焊点。但仔细观察可见到电路板上备有一小圆孔，透过小孔能看到一个可变电阻的调节长狭缝。这个可变电阻就是VR1。用螺丝刀伸进小孔可对VR1进行调节。而驱动器模拟板上的电位器R28是最易发现的。它一般被安装在IC芯片MC3470的附近，并且用封漆封住。它的作用是用来调整MC3470内部电路的工作状态。调整时应仔细操作才是。

如果R28已被调乱，可按下述步骤进行恢复调整：

1.对于大部分磁盘驱动器，可在打开主机而未执行磁盘读写命令时就直接进行调整。但对于NPH类的驱动器应在读或写磁盘命令时进行调整。

2.用逻辑笔测量IC芯片MC3470的各管脚，同时慢慢调整R28。直至使MC3470的各管脚信号如表318-1所示，则调整完毕。

表318-1 MC3470各脚正常信号

管脚号	6	7	8	10	14	15	16	17
信号	P	P	P+H	P+L	H	H	H	H

表中所示"P"表示是脉冲信号，"L"为低电平，"H"为高电平，"P+H"为脉冲的同时高电平指示发二极管也亮，"P+L"为脉冲的同时低电平指示发光二极管也亮。

319. 磁盘不转的原因有哪些

开机后磁盘驱动器不动作，磁盘不转，主机直接进入监控或BASIC状态。这类故障一般发生在驱动器接口电路或驱动器主轴电机控制电路方面。

开机后磁盘不转，驱动器工作指示灯不亮，主机直接进入BASIC状态，往往是由于驱动器电缆插件接触不良或接口电路故障所至。驱动器工作指示灯不亮说明驱动器未被加电，一般是由于驱动器接口电路没有输出ENBL信号或该信号未被驱动器电路接收到。这时一是要仔细检查插座的接触情况，再就是要重点检查U43(74LS132)和U45(NE555)

有无损坏现象。

开机后磁盘未转而主机进入监控状态。这应是驱动器启动程序 Boot0 被破坏，使程序运行到其它存储区后执行软中断"BRK"所至。这时应考虑 U35(27256EPROM)中固化的内容部分损坏。

开机后驱动器工作指示灯亮而磁盘不转，是驱动器主轴电机及其控制电路故障。这时要检查驱动器主轴电机控制电路(参见图 305-2)。首先检查 LAG570 的第 8 脚是否为低电平，若无低电平则是引线接脚故障。否则，应检查晶体管 TR2 的集电极是否有 11V 左右的电位；若没有，且晶体管 TR1 和 TR2 也没有损坏，则是 LAG570 电路故障。

320. 磁盘转速不稳的原因有哪些

磁盘转速不稳时会造成磁盘不能引导，数据读写错误，甚至会出现"毁盘"现象。造成磁盘转速不稳的原因可分为电路部分的与机械部分的。

机械部分故障主要表现为磁盘夹不紧，磁头加载弹簧过紧或磁盘质量过差。当磁盘夹不够紧时，磁盘与夹之间打滑，磁盘转速会忽快忽慢而转速不稳。为使磁头与磁盘表面接触合适，在磁头另一面与磁头对称的位置上有磁头加载机构，该机构和磁头一起将磁盘夹在中间。如果加载弹簧过紧，给磁盘运动造成较大阻力则使磁盘转速不稳。此外，如果磁盘质量较差，磁盘在磁盘套中转动受阻,也会产生此故障。

电路部分故障主要产生于主轴电机控制电路(参见图 305-2)。这时应重点检查晶体管 TR1 是否损坏或性能不良。电位器 VR1 是否接触不好，电容 C3 是否失效。否则，应怀疑 LAG570 芯片内部电路有问题或主轴电机本身有毛病。

321. 磁头不移位的原因是什么

磁头不移位会使磁盘不能引导，磁盘数据不能读写，仅磁盘转动而磁头不能沿磁盘径向移动寻道。产生磁头不移位的原因一般是步进电机驱动电路故障，而步进电机本身故障率是很低的。

驱动器步进电机是四相 12V 步进电机。四相线圈的驱动电路由高压 OC 反相器 ULN2003(D4)担任(参见附图 11)。当 IC 芯片 D4 损坏时，将不会产生步进电机各相的驱动电压，步进电机不动作，磁头也就不移位。可在开机时用逻辑笔测试 D4 的第 11,12,13,14 脚(有些驱动器改变了引脚，应预以注意)，看是否有脉冲存在。若有脉冲，则可能步进电机引线有故障。无脉冲时，则是 D4 损坏或驱动器接口上的四相时序产生电路 C2(参见附图 11)损坏。尤其是当步进电机某相信号固"0"时，磁头将固定在某一位置而不能移动。这种固"0"故障可能是 ULN2003 输出端对地短路造成的，也可能是 C2(9334 或 74LS259)输出端固"1"造成的。对于旧式的全高磁盘驱动器，其磁头移位是磁头导杆在一个带有螺旋槽沟的圆盘上沿螺旋槽沟运动而实现的。当螺旋槽沟损坏或磁头导杆跳出了槽沟，也会出现磁头不能移位的故障现象。

322. 驱动器在开机时不能引导 DOS 的原因是什么

在开机时磁盘转动，也有磁头退道的声音。但就是不能引导进 DOS(CP／M 系统也不能载入)。有时表现为磁盘不停地转而读不进数据，有时则表现为磁头反复地归零而发出磁头导杆的撞击噪声。这种故障现象往往是几台或几十台以至机房内所有驱动器在短时间内同时发生。尤其对于整个机房内的设备都是同时购置，而使用情况相同并已使用二.三年的驱动器。所以，对于机房管理人员和维修人员是一件最头痛的事。

产生这种故障的原因是由于驱动器步进电机 φA 相长期加电，而使得步进电机内转子与定子齿间相互磁化所至。读一下驱动器卡上 PROM(P5A)中所固化的磁盘引导程序(Boot0)会发现，开机时要将磁头定位于零号磁道，即磁头归零。这一动作是非常关键的，为使定位准确采取了步进电机 φA 相长时间加电的办法(其它磁道定位则是各相通电到位后随即断电)。这种长期加电是必要的，但也产生了敝端——步进电机磁化。使得当再进行第 1.2 号磁道定位时，由于磁化后的磁极牵制而不能到位。于是出现只能读第零道而不能读第 1 道，其故障现象就是反复回读第零道，发出归零时的噪声。这种故障的排除方法是将 φA 反向加电去磁(如图 322-1 所示)。去磁时加电电流可达到 200mA，每次几分钟，反复几次(注意加电极性)。

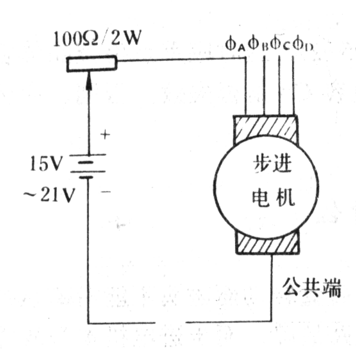


图 322-1 步进电机去磁方法

如果去磁不成功，可采取下面的方法:(但没有检修电路经验的读者请不要做此试验，否则将造成无法弥补的损失)。

1.将步进电机从固定架上取下。然后将转子与定子间相对转动 180° 再将步进机重新安装上。但是暂不要把固定螺钉拧紧。

2.找一张有 DOS3.3 系统的磁盘，放入此驱动器并开机引导 DOS。这时由于步进电机还未安装合适，磁头读不到第零道的数据。

3.慢慢地反复转动步进电机的机体，使 DOS 能被载入。这时再将步进机的固定螺钉拧紧则驱动器就可以恢复正常。

其中步骤 3 很关键，要细心地操作。有时还会反复几次才行。除了上述故障外，当磁盘驱动器的磁头或磁头读数据放大电路与读数据通路有问题时，也会产生不能引导 DOS 的情况。但是决不会产生磁头不断归零发出撞击噪声的现象。还有一种情况，就是当磁盘上第 1 磁道有错误或损坏，第零磁道无故障时，也会出现磁头不断归零的现象。

总之，开机时不能引导 DOS 的原因可有:

1.磁盘上 DOS 部分数据区损坏。(磁盘不停地转或磁头不断归零。)

2.磁头读数据放大电路故障。(磁盘不停地转。)

3.磁头太脏，数据读不进。(引导失败，进入 BASIC 状态。)

4.步进电机磁化。(磁盘不停地转或磁头不断归零。)

5.主机 RAM 故障。(DOS 引导失败。)

323. 为什么有些磁盘驱动器容易“毁盘”

磁盘驱动器“毁盘”现象常出现在三种情况下:

1.开机时载入 DOS 不成功，反而将磁盘信息破坏。即使带有写保护也无济于事。

2.向磁盘写数据时。

3.用户程序错误或键入了错误命令。这并不是磁盘驱动器电路故障故不进行分析。

开机时“毁盘”说明驱动器应该做读盘动作时却作了写盘动作。这种情况是由于驱动器模拟板电路(参见附图 11)上的 74LS125(B4 三态门)或 ULN2003(D4,高电压 OC 六反相驱动器)损坏造成的。若仅是 B4−8 或 D4−7、D4−10 端损坏，则仅对本驱动器的磁盘进行破坏。而若是 B4−15 呈高电平 B4−16 端损坏呈低电平，则在写 1 号磁盘的同时将 2 号驱动器内磁盘的数据破坏。造成这种故障的原因往往是人为的。如果在将驱动器的 20 芯电缆插头插在驱动器接口电路上时插反了方向，或只插上了一排便加电工作，则会将 B4(74LS125)和 D4(ULN2003)烧坏，造成开机时“毁盘”。

单纯的写电路故障一般不会造成“毁盘”现象。只有当读数据放大电路工作不稳定时，读磁盘磁区标志出现错误，将不应写的区域写上了数据。这就使原来磁盘的格式被破坏，再进行读盘就无法读出，即磁盘被“毁”。造成这种故障的原因一般是由于磁头读数据放大电路 B1(MC3470)的工作状态未调整好，或长时间工作后工作状态发生了变化。还有一些情况是人为造成的，比如:有些用户在调节磁盘转速时，错把为 B1 确定工作点的外接电位器 R28 当作调速电位器调乱，使得 B1 工作不稳定，从而造成“毁盘”。这种情况下，只要再将 B1 工作点调整正确就行了。调整方法可见本书有关题目。

324. 磁盘数据写不上时怎么办

造成磁盘数据写不上的原因是多方面的，它包括驱动器接口电路故障、驱动器模拟板上写请求信号通路或写数据信号通路故障及磁头电路故障。(见附图 11)

驱动器接口故障主要是写请求信号 (WR REQ)驱动电路 B2−4 (74LS05)损坏造成的。无写数据信号在不写盘故障中不能单独出现，一般是读 (RD DATA)与写数据(WR DATA)电路同时失效。

驱动器模拟板电路故障可按下述步骤进行检修:

1.让磁盘驱动器执行写动作，用示波器观察 A3 (CA3146)的 1、5 脚(有些驱动器未用 CA3146，而是用几只晶体管和电阻搭成与该集成块功能相同的差动放大电路。)看是否存在有峰值为 10V 左右的脉冲。如果有脉冲而写不上，则是磁头部分损坏。

2.如果 A3 的 1、5 脚无脉冲，则可检查 CR1、CR2 两只二极管是否开路，检查 Q2 是否损坏，还应检查 A3−8 是否为低电平。若这部分电路工作状态完好，说明磁头驱动与写数据放大电路均无故障。

3.检查 B4−9 是否有低电平存在，若没有则可能是驱动器接口或引线上有毛病。有低电平时，应检查 B4−8 是否为低，检查 D4−10 是否为高，还应检查 B4−10(写保护信号)是否为低。这些都正常时，说明写控制信号电路无故障。

4.检查 B4−5 和 B4−6 是否有数据脉冲，其中 B4−6 的脉冲幅度较小，且 B4−6 的静态直流电位在 1.6V 左右。若 B4−5 有而 B4−6 无，则应检查 B4−4 是否为低电平。若 B4−6 为低，则说明磁头写数据通路的 B4 芯片损坏。

以上分析是对 DISK Ⅱ类（或 SD−50、FD−50 等）驱动器进行的。对于非 DISK Ⅱ类驱动器，这些部分的电路是基本相同的，只是 IC 芯片管脚号有区别，检修时应注意查对。非 DISK Ⅱ类驱动器的磁头定位电路是 DISK Ⅱ类所不具备的，其发生故障的现象较少见。关于磁头定位故障检修可见专门叙述。

325. 使用打印机时主机系统被“挂住”是什么原因

首先应说明的是使用打印机时主机系统被“挂住”的根本原因为主机与打印机间的联络信号不正常。

主机与打印机间除了有 8 根数据线传送数据外，还有一根地线和三根联络线。(这三个联络信号分别为：STROBE、ACKNLG 和 BUSY。其中 STROBE 信号是主机通过打印机卡发送给打印机的“选通”信号，低电平有效。它告诉打印机，主机已将要传送的数据准备好了。ACKNLG 是打印机通过接口卡发送给主机的“回答”信号，也是低电平有效。它告诉主机，现在打印机已经接收完数据。该信号是个负脉冲，它常用于中断方式工作的接口电路，做为中断请求信号的触发。BUSY 也是打印机通过接口卡发送给主机的，但它是高电平有效，而且表示打印机正在“忙”状态，不能接收数据。该信号是个电平，它用于查询方式工作的接口电路。中华学习机的打印机接口卡中的驱动程序选择了 BUSY 信号。通过读地址 $ C1C1（ACKNLG 对应地址为 $ C1C0），检查数据位 D7 是否为“1”来判断打印机是否为“忙”状态（设打印机卡插在 1 号槽）。主机向打印机发选通信号(STROBE)是通过触动地址 $ C090 来实现的。

如此看来，造成使用打印机而系统“挂住”的原因有以下几个方面：

1、由于打印机驱动程序的存贮 ROM（EPROM2716，在打印机卡上。）损坏而使联络信号未产生。

2、由于 STROBE 信号产生电路损坏，打印机一直处于等待状态。

3、由于打印机未准备好（READY 灯不亮）或纸尽、或打印头卡住而使 BUSY 信号恒为“1”。

4、由于联络信号线的接触不良、开路、短路而造成主机呈等待状态。

5、打印机上的接口电路损坏。

以上几方面中，以打印机接口卡电路故障为最常见。而打印机的状态造成的系统“挂住”不应算是故障，纯属操作上的问题。

326. 打印机出错的原因是什么

打印机出错常指打印错误字符，打印杂乱字符，打印机不断走纸而失控等错误状态。这些错误都是因送给打印机中央控制电路的码子发生错误所至。这种错误有主机传送数据错误，有数据传输线故障造成的错码，有打印机本身接口电路故障造成的错码，有打印机

表 326-1　打印机（FX-100）ASCII 码表

n = 10：ASCII + Graphic

	UPER BIT	0	1	2	3	4	5	6	7	8	9	A	B
UNDER BIT		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011
0	0000	NUL 0	16	SP 32	0 48	@ 64	P 80	` 96	p 112	NUL 128	144	▬ 160	┴ 176
1	0001	1	DC1 17	! 33	1 49	A 65	Q 81	a 97	q 113	129	DC1 145	▬ 161	┬ 177
2	0010	2	DC2 18	" 34	2 50	B 66	R 82	b 98	r 114	130	DC2 146	▬ 162	┤ 178
3	0011	3	DC3 19	# 35	3 51	C 67	S 83	c 99	s 115	131	DC3 147	▬ 163	├ 179
4	0100	4	DC4 20	$ 36	4 52	D 68	T 84	d 100	t 116	132	DC4 148	■ 164	─ 180
5	0101	5	21	% 37	5 53	E 69	U 85	e 101	u 117	133	149	■ 165	─ 181
6	0110	6	22	& 38	6 54	F 70	V 86	f 102	v 118	134	150	■ 166	│ 182
7	0111	BEL 7	23	' 39	7 55	G 71	W 87	g 103	w 119	BEL 135	151	■ 167	│ 183
8	1000	BS 8	CAN 24	(40	8 56	H 72	X 88	h 104	x 120	BS 136	CAN 152	▏ 168	┌ 184
9	1001	HT 9	25	) 41	9 57	I 73	Y 89	i 105	y 121	HT 137	153	▎ 169	┐ 185
A	1010	LF 10	26	* 42	: 58	J 74	Z 90	j 106	z 122	LF 138	154	▎ 170	└ 186
B	1011	UT 11	ESC 27	+ 43	; 59	K 75	[91	k 107	{ 123	UT 139	ESC 155	▍ 171	┘ 187
C	1100	FF 12	28	, 44	< 60	L 76	\ 92	l 108	¦ 124	FF 140	156	▍ 172	╭ 188
D	1101	CR 13	29	- 45	= 61	M 77	] 93	m 109	} 125	CR 141	157	▌ 173	╮ 189
E	1110	SO 14	30	. 46	> 62	N 78	^ 94	n 110	~ 126	SO 142	158	▌ 174	╰ 190
F	1111	SI 15	31	47	? 63	O 79	_ 95	o 111	DEL 127	SI 143	159	+ 175	╯ 191

缓存器故障造成的错码。

主机传送错误往往是用户对打印机字符与控制码不了解，误送了一些控制码所至。这不是电路问题，故不讨论。

数据传送故障是指主机打印卡到打印机接口电路间的8条数据线中，若有个别线发生开路或对地短路而造成固"1"或固"0"错误。则如果是低数据位故障时，会产生重码错误。如：打印字符B和字符C时都打印成B(D0位固"0")或都打印成C（D0位固"1"）。从字符的ASCIIⅡ码表（表326−1）中可分析出这种故障时有问题的数据线的所在。而在高数据位故障时会产生打印机动作错误。如：打印字符L时却使打印机走纸（D6位固"0"）。这种故障也可从ASCII码表中分析出来。

打印机本身接口电路故障常见于数据锁存电路74LS374个别位损坏所至。这时，打印机电缆上数据和信号都正常，但打印机仍出错。通过检查74LS374的数据输入、输出端可发现故障点。

打印机缓存器故障可通过改变打印机上工作状态设置开关来发现（详见打印机使用说明）。

327. 打印机打印出字符缺一行点是何原因

打印字符时缺一行点往往是在一些使用时间长的打印机上有发生。产生这种现象的原因有如下几个方面：

1、打印机使用时间过长，且一直未换色带，色带上的色带油已用光。打印时打印头与打印纸间的距离调整不合适。这时往往在字符的最上一行或最下一行出现缺点现象。本故障可通过为色带加色带油或更换新色带的办法解决。

2、打印机头内打印针断或打印机使用时间过长而使打印机针头磨秃，此时，打印的字符缺行现象可发生在字符中间位置，也可出现于上下两头。如果仅是针头磨秃，可通过调整打印头与打印纸的距离来排除故障。而如果是打印针断了，则只能通过换打印头来解决。

3、打印头内打印针驱动线圈断路。发生此故障时现象与2相同。测量打印针驱动线圈可发现开路现象。这时可将打印头打开，找到有故障的线圈。如果是接线端开路可将其焊好，若是线圈内部断线则只能更换打印头。因打印针驱动线圈都是用树脂封装的，不能拆下。

4、打印针线圈的驱动晶体管损坏。故障现象同3。在排除其它故障可能性时可将打印机壳打开，在电路板上可见到有一排9个大功率晶体三极管。找到缺行针所对应的那只晶体管，测量其工作状态或将其拆下测量，可发现晶体管已损坏。更换晶体管便可排除故障。

5、打印机电路板到打印机头间的连线，由于长期运动使内部断线。故障现象同3。这可通过测量发现断线部位，然后将其接上即可。

328. 怎样给打印机色带上色带油

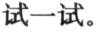
打印机长期使用后，色带的颜色会变淡，打印出的字符不清楚。更换色带当然是个最简单的办法。但有时买不到适用的色带。如果打印机色带并没有破损，则可通过加色带油的办法来恢复打印效果。

买一瓶色带油，找一、两块脱脂棉或泡沫塑料。将色带拉出一段铺平到一张纸上。用棉花或泡沫塑料沾少许色带油，然后在色带上反复涂抹。注意一定不要太多。涂抹完毕可先打印几个字符看看浓淡是否合适。合适后就可将色带逐渐拉出，分段上油，再转回去，直至整条色带上油完毕。

329. 怎样使两台主机共用一台打印机

通常中华学习机的配置为一台主机带一台磁盘驱动器和一台打印机。但是由于打印机价格较高（与主机价格相当），而使用的时间又远远小于主机和磁盘驱动器，所以许多用户在购置中华学习机时往往不配置打印机。这就造成了一个矛盾：在一个计算机房中，接有打印机的主机并不是经常使用打印机；恰恰相反，打印机是经常闲置着的。而未接打印机的主机操作者一旦想使用打印机时又得将程序存到盘上，再到有打印机的主机上将程序调出运行或列表打印。这显然是非常麻烦的。如果能将一台打印机接到两台主机上，将会带来不少方便。办法如下：

给无打印机的主机配置一个打印卡（可用本书介绍的中华学习机打印卡）如图 329−1 所示将两台主机的打印机卡输出线经一分线开关后连至打印机的输入端。当某台主机需要用打印机时将开关拨向该主机一边即可使用。另一台主机若此时也想用打印机，则当键入“PR#1”后，该主机系统被“挂住”；直到将开关再拨过来或按“CTRL−RESET”键后才能恢复。这虽然不是一个最好的办法，但简单易行，价格又便宜，有兴趣的读者及用户不妨试一试。



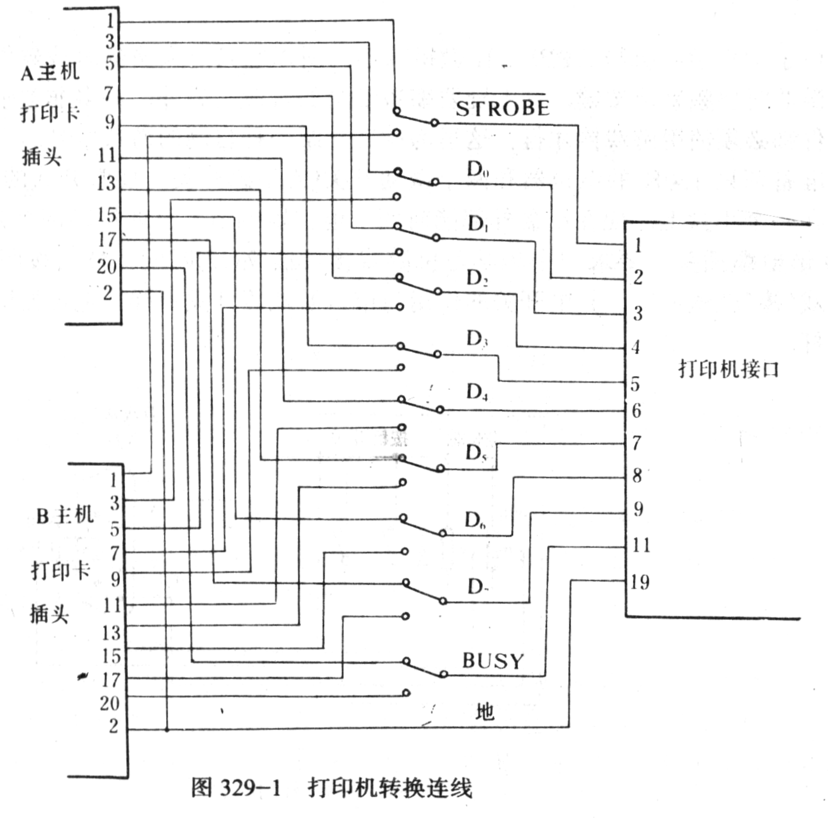


图 329−1　打印机转换连线

330. 怎样实现多台主机脱机共享一台打印机

在上一个题目中我们介绍了一种两台主机共用一台打印机的简法。这一方法的缺点是使用打印机时要用户用手去拨动开关，不是很方便。更大的问题是当一台主机打印时，另一台主机若也想打印就只好停下来等着对方打印完。如果要是能多台主机共享一台打印机，并且打印方式又是脱机形式的，那将给用户带来更大方便。北京师范学院分院校办厂(双新电子设备厂)生产的 MPC－Ⅱ可同时接四台主机和一台打印机，并且对主机的机型(8位机和 16 位机均可)及打印机的机型（9 针的和 24 针的均可)没有限制。只要使用 CENTRONIC 标准并行接口就行。当四台主机同时使用打印机时，四台主机要打印的内容很快一起被送到 MPC－Ⅱ的内存贮器中。然后各主机就可以去干别的事情，而打印的事情就由 MPC－Ⅱ来承担。它会按顺序将各主机的内容分别打印出来。

不仅如此，MPC－Ⅱ还可以将一台主机与四台并行口的外设(打印机、绘图仪等)相接。通过手控选定 1 至 4 台不同外设或程控选定 4 台外设中之某一台，将主机的数据送到外设中去。另外，MPC－Ⅱ还可以由主机送入它的各种不同控制程序，以实现各种不同要求的功能。MPC－Ⅱ可算是一台性能完善的微机接口控制器。若想更多了解关于它的一些情况，可与该机的生产厂家联系

331. 怎样自制游戏棒

由于中华学习机与 APPLE Ⅱ 微机兼容，所以它具有大量的游戏软件。但是由于玩计算机游戏时要频繁地按键，这无疑要缩短键盘的寿命。另外，许多非常吸引人的游戏与绘图软件都必须使用游戏棒才行。这里为读者介绍一种自制的游戏“棒”。

准备两只 150K 的电位器和两个微动开关(或手感开关，要常开式的)。将这四个元件固定在一个小盒上，使在小盒外部能对两个电位器调节，且能按动两个开关。再买一个 9 芯 D 形电缆插头，要求与中华学习机游戏接口插座(J2)匹配。然后按图 331-1 接线，一个游戏“棒”就做成了。其中开关的作用与市售游戏棒开关一样，电位器则代替市售游戏棒的摇杆。

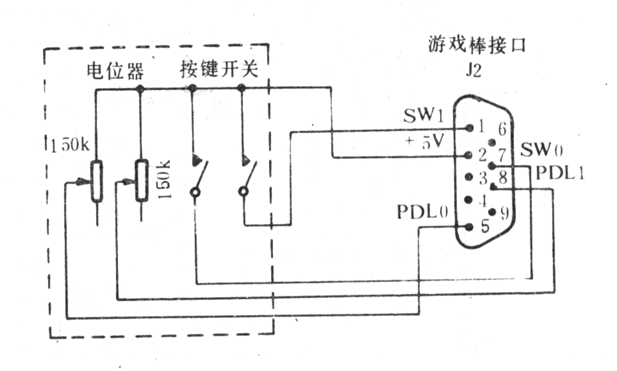


图 331-1 游戏“棒”接线图

332. 怎样使中华学习机的扬声器发声更大

中华学习机一般扬声器发声都比较小。若希望其发声再大些，可对电路进行小小的改动。办法之一是在主板的扬声器插座(J8)附近找到电阻 R28 (阻值 75Ω。见附图 3)。在该电阻上再并联一只 50Ω 左右的电阻，便可使扬声器发声变大。但这样做后改变达林顿晶体管 Q3 的负载电阻很多，造成功率失配。所以比较好的办法是将原电路进行改造，成为如图 332-1 所示的样子。这样做要对主板电路改接，工程大，但效果好。

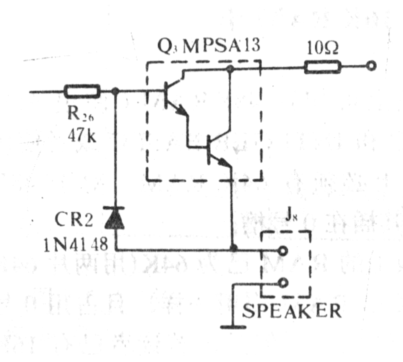


图 332-1　扬声器电路的改接

再有一种办法是将主机产生的声音信号先调制到 6.5MHz 的信号上，再送到调制器里调制到射频信号上，由电视机的扬声器来发声。但这种电路在业余条件下已不易制作，可去购买这类射频调制器。

333. 中华学习机有哪些常用接口卡

目前，世界上流行的 APPLE Ⅱ微机接口卡多达上千种。在国内各类用途的接口卡也有数十种之多。这些接口卡都可用于中华学习机。下面我们就以扩展 I／O 接口设定的槽号为序，按接口卡习惯使用插槽将各种接口卡分类简述。

1.第 1 号槽：常插入打印机接口卡。

(1)EPSON 打印机卡。此卡使用流行最广。为标准 CENTRONIC 并行接口，可用于各种该类接口的 9 针(80 列或 132 列)打印机。此卡有打印图形的功能，所以可用于汉字的打印。在使用有些版本的 CP／M 操作系统时(如 CP／M2.2)，由于系统不认识此卡，故不支持打印。但使用 CP／M2.2B 或改变 CP／M 升级程序后，可支持此卡。

(2)打印缓冲卡。此卡上有 32K 或 64K 的数据缓冲 RAM，使用时主机可实现脱机打印。

(3)中华学习机汉字打印卡。此卡售价极低。请详见本书有关问题。

2. 第 2 号槽：常使用 EPROM 写入卡，串行接口卡或 AD／DA 变换卡。

(1)EPROM 写入卡可读写 2708、2716、2732、2764、27128 等 EPROM。其使用与改造可详见有关题目。

(2)串行接口卡为标准的 RS--232 接口，可用于串行数据的通迅和微机联网。

(3)AD／DA 卡用于对模拟量取样和量化及将数字量模拟化。这使得中华学习机可进行工业过程的自动化控制。

3.第 4 号槽：常使用 CPU 卡。

(1)Z80 卡。上有 Z80 CPU，可支持 CP／M 操作系统，运行该系统下的各种高级语言程序(BASIC、FORTRAN、编译 BASIC 等)和关系数据库(d-BASE Ⅱ)。此外，还可运行 Z80 或 8080 汇编语言程序。为在中华学习机上开发 Z80 系列的各种应用软件创造了硬件环境。

(2)8088 卡。插上该卡后，中华学习机就成为一台 16 位微机，可运行 8088 汇编程序。

4.第 5 号槽：常使用各种其它接口卡。如：IEEE–488 接口卡，发声卡，时钟卡，日历卡等。

334. 中华学习机为什么不需要 16K RAM 卡

早期的 APPLE II 和 APPLE II⁺微机最多在主板上可配置 48KRAM（使用 4116 RAM 芯片）。如果希望同时使用 APPLE SOFT BASIC 和 INTEGER BASI C 或者使用 56K CP／M 操作系统、Pascal 操作系统等，则主板上必须有 64K RAM。对于 48K RAM 的微机于是就要另配一个 16 KRAM 扩展卡，将其插在 0 号槽。

中华学习机为 APPLE IIe 微型机的兼容机，其主板上的 RAM 已为 64K(用两片 64K X 4 位的动态 RAM 50464)。其地址最高的 16K 与 16K 卡 RAM 作用一样，且占用 0 号槽。所以中华学习机就不再需要 16K RAM 卡扩展存储区了。但是，若读者已有 16K RAM 卡，也可用于中华学习机，方法参见有关题目。

335. 中华学习机如何使用 16K RAM 卡

中华学习机主板上已做好 64K RAM，如果读者手头有多余的 APPLE II 微机 16K 扩展 RAM 卡，可用本书介绍的方法将其利用起来。但是，要求该 16K 卡是不带 16 脚插头引线的那一种。

将 16K 卡插在中华学习机扩展插槽上，使用时按表 335–1 所列地址去触动它们，便可读写新插上去的 16K 卡上的扩展 RAM。

表 335–1　16K RAM 卡选用方式触动地址

存储器分区	8KRAM 体和 4KRAM 体 A	8KRAM 体和 4KRAM 体 B	扩展卡 RAM 与监控 ROM 被选中情况
触动地址及功能	触动 $ Cono	触动 $ Con8	监控 ROM 不可读，但扩展 RAM 可读、不可写
	触动一次 $ Con1 触动两次 $ Con1	触动一次 $ Con9 触动两次 $ Con9	监控 ROM 可读，扩展 RAM 不可读 监控 ROM 可读，扩展 RAM 可写
	触动 $ Con2	触动 $ ConA	监控 ROM 可读，扩展 RAM 不可读、不可写
	触动一次 $ Con3 触动两次 $ Con3	触动一次 $ ConB 触动两次 $ ConB	监控 ROM 不可读，扩展 RAM 可读， 监控 ROM 不可读，扩展 RAM 可读、可写

表中 n = 槽号 + 8。例如：若 16K RAM 卡被设定在 2 号扩展槽，则 n = 2+8 = 10 = $0A。下列各指令及功能为：

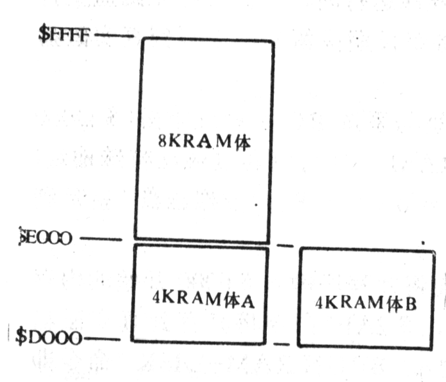


图 335-1 16K 扩展 RAM 的地址分配

LDA $C0A9;(第一次触动)监控 ROM 可读，扩展 RAM 不可读。

STA $C0A9;(第二次触动)监控 ROM 可读，扩展 RAM 可写。

IT $C0AA;(仅一次触动)监控 ROM 可读，扩展 RAM 禁读禁写。

利用这类指令，可将一些内容存放到 16K 扩展 RAM 中以便各种处理。也可实现"虚拟磁盘"。

表中的 4K RAM 体 A 和 4k RAM 体 B，是指将 16K 卡上的 16K RAM 分成三块，然后构成两种组合管理(参见图 335-1)。这三块分别是 8KRAM 体(占地址 $E000～$FFFF)、4K RAM 体 A 和 4K RAM 体 B(都占地址 $D000～$DFFF)。由于两个 4K RAM 体地址相重叠，所以要分别与 8K RAM 体结合使用。

336. 怎样快速检测 16K RAM 卡

中华学习机插上 16K RAM 卡之后可扩充许多功能。但在使用之前要想先测试一下该 16K RAM 卡，可采用本文介绍的方法。检测是通触动表 335-1 所列的一些 I／O 地址来实现的。操作如下：(在监控状态下进行)

```
*C081                (1. ROM可读)
*C081                (2.读监控 ROM写 8K RAM + 4K RAM体 A)
*D000<D000.FFFFM     (3.将 ROM内容搬入扩展 RAM)
*C083                (4.读扩展 RAM)
*E000G               (5.进入 BASIC状态)
]CALL -151
*C089                (6.开始检测 8K RAM + 4K RAM体 B)
*D000<D000.FFFFM
*C08B
*E000G
]
```

若在第 4 步时"*"不出现，则说明该 16K RAM 卡没有插好或损坏，或槽号不对。而若在第 5 步后不进入 BASIC 状态，则说明该 16K 卡上的 RAM 有损坏的。

这种检测方法又迅速又可靠，有兴趣的读者不妨一试。

337. 中华学习机如何实现"虚拟磁盘"

DOS 是用于控制主机与磁盘驱动器之间进行数据交换的程序。如果将其改造后便可

用于管理主机 RAM，把主机 RAM 当作磁盘使用。这种将内存储器当作磁盘的方式称"虚拟磁盘"。其优点是存取速度快(大约是真实磁盘存取速度的 40 倍)，而且用于真实磁盘的 DOS 命令全部适用于"虚拟磁盘"。在较大的实用程序运行时，凡需要建立磁盘文件的中间结果可暂时写到"虚拟磁盘"上，程序结束前再将要长期保留的结果写到真实磁盘上。这样处理后可使程序的运行速度加快几十倍。

中华学习机主机内存储器为 64K，其中有 16K 地址与系统 ROM 地址重叠($ D000 ~ $ FFFFF)。这 16K RAM 与 16K RAM 卡的扩展 RAM 一样可以通过触动特殊地址(见表 335-1)来进行管理，只是其所占槽号为 0 号(即 n = 0 + 8 = 8)。"虚拟磁盘"就是利用这 16K RAM 来实现的。

制做"虚拟磁盘"的程序列于表 337-1 中。读者可先将此程序键入 $ 6000 开始的内存当中，检查无误后用" BSAVE RAM-DISK,A $ 6000,L $ 2AF"命令将其存到真实磁盘上。当希望使用"虚拟磁盘"时，只要在载入 DOS 后键入" BRUN RAM-DISK "命令即可。

此后的 DOS 命令将是在"虚拟磁盘"上进行管理操作。

表 337-1 RAM-DISK 程序的机器码

6000-	A0	02	B9	55	60	99	B7	B7	6150-	A9	04	0A	0A	18	69	01	85
6008-	88	10	F7	A0	02	B9	58	60	6158-	1F	AD	ED	9C	C9	01	D0	04
6010-	99	2A	BD	88	10	F7	A9	60	6160-	A9	00	F0	03	AD	E0	9C	8D
6018-	8D	6B	A5	A9	0E	8D	93	B2	6168-	E4	9C	2C	A6	9A	10	17	A5
6020-	A9	5B	85	3C	A9	60	85	3D	6170-	1F	8D	EC	B7	20	04	BD	B0
6028-	A9	A9	85	3E	A9	62	85	3F	6178-	76	A0	05	B9	BB	B3	D9	DA
6030-	A9	A6	85	42	A9	9A	85	43	6180-	9C	D0	60	88	D0	F5	AD	E9
6038-	A0	00	20	2C	FE	A9	02	8D	6188-	B7	CD	BB	9C	F0	55	8D	E2
6040-	57	AA	A9	80	8D	00	9D	A9	6190-	9C	AD	EA	B7	8D	E3	9C	A4
6048-	9A	8D	01	9D	20	D4	A7	20	6198-	1F	C8	8C	E5	9C	A9	00	8D
6050-	53	9C	4C	7E	9B	20	A7	9A	61A0-	E9	9C	A9	E1	85	48	A9	9C
6058-	20	AF	9C	FF	84	48	85	49	61A8-	85	49	A9	03	85	1E	A4	1E
6060-	A0	1D	A2	19	A9	A0	D9	75	61B0-	BE	BC	9C	BD	83	C0	BD	83
6068-	AA	D0	2C	88	CA	10	F7	A0	61B8-	C0	B9	C0	9C	09	0F	8D	EA
6070-	02	B9	B8	9C	D9	75	AA	D0	61C0-	9C	A9	0F	8D	E6	9C	20	04
6078-	1E	88	10	F5	AC	5F	AA	C0	61C8-	BD	B0	24	CE	EA	9C	CE	E6
6080-	0C	F0	0B	C0	0E	F0	07	C0	61D0-	9C	10	F3	CE	E5	9C	C6	1E
6088-	05	B0	0C	4C	5B	9B	98	E9	61D8-	10	D4	AD	EF	9C	8D	E0	9C
6090-	0E	8D	A6	9A	4C	7E	9B	AD	61E0-	4C	34	9C	2C	82	C0	20	3A
6098-	E9	B7	CD	BB	9C	D0	6E	AD	61E8-	FF	2C	82	C0	4C	7E	9B	2C
60A0-	F4	B7	F0	58	C9	04	D0	03	61F0-	82	C0	A0	0D	B1	48	4A	4A
60A8-	4C	7B	9B	48	AD	EC	B7	C9	61F8-	4A	4A	A0	03	88	F0	03	4A
60B0-	13	90	04	E9	04	B0	F8	C9	6200-	90	FA	B9	F2	9C	4C	85	B3
60B8-	0F	B0	04	69	04	90	F8	8D	6208-	A9	FF	85	47	A0	00	84	46
60C0-	78	04	E9	0F	AA	AD	F0	B7	6210-	98	2C	83	C0	2C	83	C0	91

```
60C8-  85  3E  AD  F1  B7  85  3F  BC
60D0-  BC  9C  B9  83  C0  B9  83  C0
60D8-  BD  C0  9C  0D  ED  B7  85  47
60E0-  A9  00  85  46  68  4A  B0  0B
60E8-  A0  00  B1  3E  91  46  88  D0
60F0-  F9  F0  09  A0  00  B1  46  91
60F8-  3E  88  D0  F9  2C  82  C0  AD
6100-  BB  9C  8D  F7  B7  AD  E0  9C
6108-  8D  F6  B7  18  60  4C  04  BD
6110-  98  D0  2A  A9  10  2C  65  AA
6118-  D0  04  A9  50  D0  03  AD  E9
6120-  B7  8D  BB  9C  AD  EB  B7  C9
6128-  FE  90  02  A9  00  8D  E0  9C
6130-  20  53  9C  A9  00  85  48  20
6138-  EA  A2  4C  8B  9F  4A  8D  ED
6140-  9C  AD  78  AA  C9  B1  90  08
6148-  C9  B9  B0  04  29  0F  D0  02
6218-  46  2C  8B  C0  91  46  88  D0
6220-  F3  C6  47  A5  47  C9  D0  B0
6228-  E7  A0  0D  B9  C4  9C  99  74
6230-  E0  88  10  F7  A0  07  B9  D2
6238-  9C  99  30  E0  88  10  F7  A9
6240-  7A  8D  27  E0  A0  06  B9  DA
6248-  9C  99  00  E0  88  10  F7  A9
6250-  11  A0  02  8D  01  E3  8C  02
6258-  E3  88  8D  01  E2  8C  02  E2
6260-  2C  82  C0  60  EC  BB  9C  F0
6268-  03  DD  8C  C0  60  D2  AD  C4
6270-  50  00  08  08  08  D0  D0  E0
6278-  F0  FF  FF  00  00  FF  FF  00
6280-  00  FF  F0  00  00  FF  FF  11
6288-  01  00  00  23  10  00  01  04
6290-  11  03  D2  AD  C4  D4  01  60
6298-  01  00  11  00  FB  B7  00  00
```

在这里，“虚拟磁盘”被假设在第 5 号槽上。所以当需要真实磁盘操作时，要用诸如“CATALOG，S6”之类的 DOS 命令；而对“虚拟磁盘 ”要用“CATALOG，S5”之类的命令。

338. 怎样制作用于存放“虚拟磁盘”文件的磁盘

上一个题目中我们介绍了如何利用 16K 扩展 RAM 做“虚拟磁盘”。其中 RAM-DISK 是一个修改 DOS 的程序。经修改的 DOS 仍然常驻 $ 9600～ $ BFFF 的 RAM 区，并能同时管理“虚拟磁盘”与真实磁盘。如果读者望能将“虚拟磁盘”的 16K 内容整块地存到真实磁盘上（不管是否写满文件），则可运行下面的 BASIC 程序来初始化一张磁盘：

```
10  TEXT : HOME
20  IF ( PEEK (47032) <  > 167) THEN PRINT  CHR$ (4);"BRUN RAM-DISK"
30  PRINT : FLASH : PRINT "WARNING: ";: NORMAL : PRINT "THIS PROGRAM
WILL DESTROY ": PRINT "          ANYTHING ON THE DISK!!!"
40  PRINT "PUT AN EMPTY DISK INTO THE DRIVE ..."
50  PRINT "IF OK HIT RETURN": GET A$: PRINT : IF A$ < > CHR$ (13) THEN
GOTO 80
60  D$ = CHR$ (4): PRINT D$;"INIT R-D": PRINT D$:"INIT HELLO,S6":
PRINT D$;"UNLOCK R-D"
70  FOR E = 1 TO 8: PRINT D$;"SAVE R-D";E: NEXT E: PRINT D$;"LOCK R-D":
HOME: VTAB 8:PRINT "PREPARE ANOTHER DISK ?": GET A$: IF A$ = CHR$ (89)
THEN GOTO 40
80  HOME : PRINT "GOODBYE ...": END
```

初始化后的磁盘上共有 8 个“虚拟磁盘”区，每个区对应着一个“虚拟磁盘”的全部空间。这 8 个区的磁盘分配情况见表 338-1。各区的名称分别为“R-D1”~“R-D8”，需要将“虚拟磁盘”内容存入某一区时可键入命令“ SAVE R-Dn,S6” (其中 n 为区号，可取值 1~8)。

表 338-1 “虚拟磁盘”在真实磁盘上的各区分配

文件名称	在磁盘上所占磁道	目录磁区所在磁道
DOS	0~2	
R-D1	3~6	5($05)
R-D2	7~10	9($09)
R-D3	11~14	13($0D)
R-D4(R-D)	15~18	17($11)
R-D5	19~22	21($15)
R-D6	23~26	25($19)
R-D7	27~30	29($1D)
R-D8	31~34	33($21)

为了让读者在使用该磁盘与“虚拟磁盘”时方便，我们在这里列出一个 BASIC 程序。运行该程序后，真实磁盘上的“虚拟磁盘”内容会顺序调入 16K RAM 卡内，并被“ CATALOG ”出来。如果读者想使用该内容，则按“ Y ”，否则按其它键。该程序名为“ CATALOG ALL ”：

```
5   TEXT : HOME
10  D$ = CHR$ (4)
20  FOR I = 1 TO 8
30  PRINT D$;"LOAD R-D";I;",S6"
35  PRINT "     R-D";I
40  PRINT D$;"CATALOG,S5"
50  VTAB 22: PRINT "DO YOU WANT IT (Y/N)?": GET A$
60  IF A$ = "Y" THEN END
70  PRINT D$: PRINT
80  HOME
90  NEXT I
```

另外，用我们前面所列初始化程序初始化的磁盘上没有 DOS，所以必须用 COPY 程序将正常 DOS 盘的第 0 至第 2 磁道内容考贝到初始化的磁盘上。并且在 R-D4 (或 R-D 区内存上程序“RAM-DISK”和“CR-D.OBJ” (见下题)。最后再将程序“ CR-D ” (见下题中的 BASIC 程序) 以“ HELLO ”的文件名也存到磁盘的 R-D4 区内。以后再使用该磁盘，开机后会自动运行“ CR-D ”程序 (现名为“ HELLO ”程序)，并向用户提问 16K RAM 卡所在槽号。回答后便自动修改 DOS，建立“虚拟磁盘”。

339. 怎样用 16K RAM 卡做“虚拟磁盘”

如果读者手头有多余的 16K RAM 卡也可用作“虚拟磁盘”。该卡插在扩展槽上后可设定为 1、2、4、5、7 号槽。然后将表 339-1 所列程序机器码按标注地址（$ 300 开始)键入，再用 DOS 命令“ BASAVE CR-D.OBJ,A $ 300,L $ 40 ”存到真实磁盘上。

表 339-1　CR-D.OBJ 程序的机器码

```
0300-  AD  82  03  0A  0A  0A  0A  69
0308-  82  8D  FD  60  8D  E4  61  8D
0310-  EA  61  8D  F0  61  8D  61  62
0318-  69  01  8D  D3  60  8D  D6  60
0320-  8D  B4  61  8D  B7  61  8D  12
0328-  62  8D  15  62  69  08  8D  1A
0330-  62  4C  00  60  00  00  00  00
```

CR-D.OBJ 程序存入磁盘后再键入下列 BASIC 程序，也存到磁盘上：(程序名为 CR-D)

```
10  TEXT : HOME
20  PRINT  CHR$ (4);"BLOAD RAM-DISK"
30  PRINT
40  INPUT  "INPUT SLOT NUMBER OF 16K CARD:";A
45  IF A < 0 OR A > 7 THEN 40
50  POKE 898,A
60  PRINT  CHR$ (4);"BRUN CR-D.OBJ"
```

运行该程序，(程序运行时要提问准备做“虚拟磁盘”的 16K 卡所在槽号，应如实输入)。则 I／O 扩展卡的扩展 RAM 就成为“虚拟磁盘”了。(程序运行时，磁盘上必须有第 337 题所述程序 RAM-DISK 和程序 CR-D.OBJ 。)

340. 怎样正确使用 EPROM 写入卡

在中华学习机和 APPLE Ⅱ微机上常用的 EPROM 写入卡型号为 AP-64E。这种写入卡可以读写 2716、2732、2764 三种不同容量的 EPROM。下面将其使用时的注意事项与操作过程叙述一下。

(1)EPROM 的正确装入：在 AP-64 卡上装有一个 28 脚的活动插座。当读写 2716 与 2732 时仅使用 28 个脚中的 24 个。卡上的集成电路块除了写有程序的 EPROM(2716)外，所有的集成块安装方向都是一致的。即集成块的缺口向左。被读写的 EPROM 芯片在装入时应与卡上大多数芯片安装方向一致，缺口向左。且对于 2716 和 2732，在装入活动插座时，应向右靠，使用活动插座上最右边的 24 个脚。这点要特别注意，一旦装反，芯片将被永久性损坏。

(2)写入卡上选择开关的设置：在 AP-64E 卡上有 8 个一排的小开关。读写不同的 EPROM 时开关放置的位置是不同的，具体位置详见表 340-1。开关拨向上为开(ON)，拨向下为关(OFF)。表 340-1 中用"0"表示"ON"，用"X"表示 OFF"。

表 340-1 选则开关位置

开关位置	1 2 3 4 5 6 7 8
2716	o o o x x x x x
2732	x x x o o o x x
2764	x x x x o o o o

(3)AP-64E 操作步骤：

①按上述(1)和(2)将要读写的 EPROM 装入并设置好开关位置。

②将 AP-64E 插入中华学习机的接口插槽上，并可任意设定槽号。

③打开中华学习机电源。若在 CEC-BASIC 状态下，可敲入"PR#s"("S"为 AP-64E 所在槽号)。若在监控(提示符"＊")状态下，可敲入"s CTRL-P"。荧光屏上将显示：

```
AP-64 EPROM PROGRAMER
2)2716
4)2732
8)2764
?
```

④根据放入 AP-64E 中 EPROM 的型号，分别敲入 2、4 或 8。荧光屏上将显示：

```
1)WRITE
2)READ
3)COPY
4)COMPARE
5)BLANKCHECK
6)MONITOR
?
```

敲入 1～6 分别为写 EPROM、读 EPROM、考贝(复制)EPROM、比较两个内容是否相同、检查 EPROM 是否是空的和进入监控状态。

⑤如果敲入"1"，则为写 EPROM。荧光屏显示："START ADDRESS?"这是问你将要写入 EPROM 内容在内存中的首地址。应该用十六进制数来回答。接着荧光屏显示"BLANK?(Y／N)"。这是问你 EPROM 是否是空的(没有写过内容)。如果你想要写这个 EPROM，则敲入"Y"。这个 EPROM 可写，则显示"ROM CHECK OK !"。否则显示"ROM CHECK ERR!!"，表示这个 EPROM 已被写入内容。

⑥如果这个 EPROM 可写，则除了显示" ROM CHECK OK!"外，还显示"SW ON"。意思是让你将 AP-64E 卡上发光二极管旁的小开关打开。这个开关打开后，发光二极管发亮，并且自激产生的 21V 电源将加到欲写入内容的 EPROM 的 Vpp 端。

⑦经上述操作后，只要按回车键，AP-64E 卡自动将内容写到 EPROM 内。写完之

后显示“COMPARE OK!!”，表示写入的内容与原内容比较后一致。若显示“COMPARE ERR!!”，则表示写错了或EPROM损坏了。除上述两种显示外，荧光屏上还要显示“SW OFF”，提示你将21V电压关掉。

⑧其它功能(如：读、考贝、比较等)也都是在一系列提示之下进行的，这里不再过多叙述。

最后需要提醒一点：无论是插、拔AP-64E卡，都先要将主机电源关掉后进行。

341. 如何用EPROM写入卡AP-64E读写27256EPROM

中华学习机上使用的EPROM写入卡AP-64E最多只能读写2764EPROM。若想用该卡读写更大容量的EPROM，如：27256，则必须对该卡的软、硬件进行改造。下面我们介绍一种改造AP-64E的方法，使其能读写27256。

1.硬件改造：首先将28脚活动插座的第26脚与28脚之间电路板连线断开，然后按图341-1所示接线。增加的1×2小开关K可焊在卡上一个合适的位置(如：选择开关的旁边)。

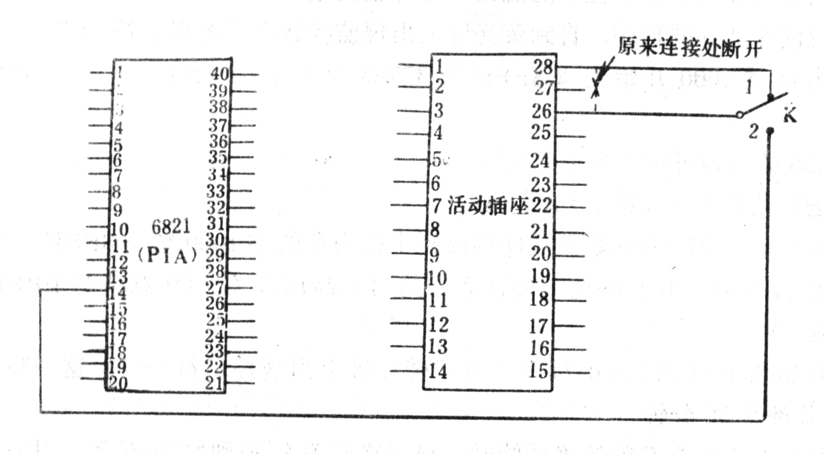


图341-1 AP-64E改造接线

如果需要写27256，则应注意27256芯片上所注写入电压。一般该电压为12.5V。所以，还需将一支12.5V的稳压管并接在AP-64E卡上稳压二极管D2（21V）两端。

2.读、写程序：如果直接使用AP-64E上的EPROM之中的程序来读、写27256是不行的。所以要重新编写程序。下面列出的程序读者可从键盘敲入后存到磁盘上，以便读、写27256时使用。

表341-1 读27256程序

0300-	A9	00	8D	A1	C0	8D	A3	C0
0308-	8D	A0	C0	A9	FF	8D	A2	C0
0310-	A9	04	8D	A1	C0	8D	A3	C0
0318-	A9	02	8D	A2	C0	A9	00	85
0320-	F0	A9	10	85	F1	A2	05	A0
0328-	00	20	70	03	C8	D0	FA	E6
0330-	F1	A5	F1	C9	30	D0	F2	A2

表341-2 写27256程序

0300-	A9	00	8D	A1	C0	8D	A3	C0
0308-	A9	FF	8D	A0	C0	8D	A2	C0
0310-	A9	04	8D	A1	C0	8D	A3	C0
0318-	A9	02	8D	A2	C0	A9	00	85
0320-	F0	A9	10	85	F1	A2	05	A0
0328-	00	20	6B	03	C8	D0	FA	E6
0330-	F1	A5	F1	C9	30	D0	F2	A2

0338-	15	20	70	03	C8	D0	FA	E6	0338-	15	20	6B	03	C8	D0	FA	E6
0340-	F1	A5	F1	C9	50	D0	F2	A2	0340-	F1	A5	F1	C9	50	D0	F2	A2
0348-	01	20	70	03	C8	D0	FA	E6	0348-	01	20	6B	03	C8	D0	FA	E6
0350-	F1	A5	F1	C9	70	D0	F2	A2	0350-	F1	A5	F1	C9	70	D0	F2	A2
0358-	11	20	70	03	C8	D0	FA	E6	0358-	11	20	6B	03	C8	D0	FA	E6
0360-	F1	A5	F1	C9	90	D0	F2	4C	0360-	F1	A5	F1	C9	90	D0	F2	4C
0368-	69	FF	00	00	00	00	00	00	0368-	69	FF	00	CA	8E	A2	C0	B1
0370-	CA	8E	A2	C0	AD	A0	C0	91	0370-	F0	8D	A0	C0	A9	90	20	A8
0378-	F0	E8	8E	A2	C0	60	00	00	0378-	FC	E8	8E	A2	C0	60	00	00
0380-	FF								0380-	FF							

3.操作步骤：读 27256 的过程按下列步骤操作：

(1)将 AP-64E 上新增设的开关 K 扳向“2”的位置。AP-64E 上原来的八个选择开关中，K2、K3、K5、K7 扳到“ON”的位置；其余四个在“OFF”位置。

(2)将待读的 27256 EPROM 放入活动插座固定好，然后将 AP--64E 插到中华学习机的扩展插槽上（设定为 2 号槽）。

(3)插入存有读写 27256 程序的磁盘，打开主机电源。

(4)运行读 27256 的程序，直到荧光屏上出现监控状态下的提示符“ * ”。

(5)从内存 $ 1000 开始至 $ 8FFF 的 32768 个单元的内容即为 27256 EPROM 的内容。

写 27256 的过程可按下列步骤操作：

(1)先进行读 27256 的第 (1)至第 (3)步。

(2)将需要写入 27256 的数据顺序存放到主机内存的 $ 1000 至 $ 8FFF 一段内。

(3)打开 AP-64E 卡上发光二极管旁边的写入高压开关。(注意待写 EPROM 上所注写入电压值。)

(4)运行磁盘上写 27256 的程序，直至荧光屏上出现提示符“ * ”。这一写入过程大约要历时 27 分钟至 30 分钟。

当需要恢复 AP-64E 的原来功能时，只需将开关 K 扳到“1”的位置，其它操作与原来相同。

342. 如何方便地使用 80 列卡显示文本，又可以显示高分辨率图形

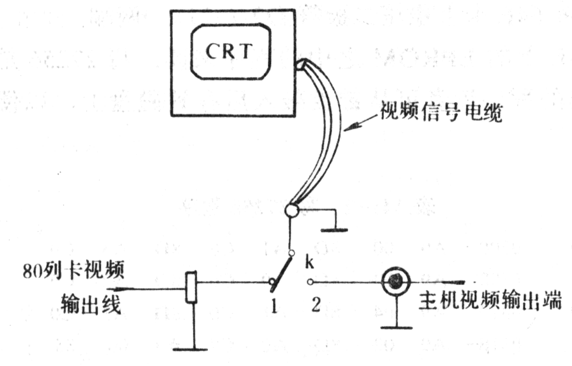


图 342-1 手动变换显示方式接线图

中华学习机上使用 80 列卡时，可将原来显示 40×24 个字符的屏幕变成 80×24 个字符的屏幕。但是，该卡只支持文本显示方式，而不支持高分辨率作图方式。(具有 64K 扩展 RAM 的 80 列卡可提供 560×192 点的高分辨率图形功能。)如果读者希望自己的微机不必改变视频信号电缆线的插接位置，便可实现 80 列文本与高分辨图形的显示变换，可对电路进行一个小小的改接。方法很简单。元件则只要有一只双向开关即可。按照

图 342-1 所示电路接线。当调试程序时，将开关 K 拨向“1”，可用 80 列的文本方式显示源程序。若程序是作图程序，运行时可将开关 K 拨向“2”，则屏幕显示高分辨率图形。

如果读者希望能在程序运行中自动地进行两种显示方式的切换，就需要做一个稍微复杂些的电路。该电路如图 342-2 所示。

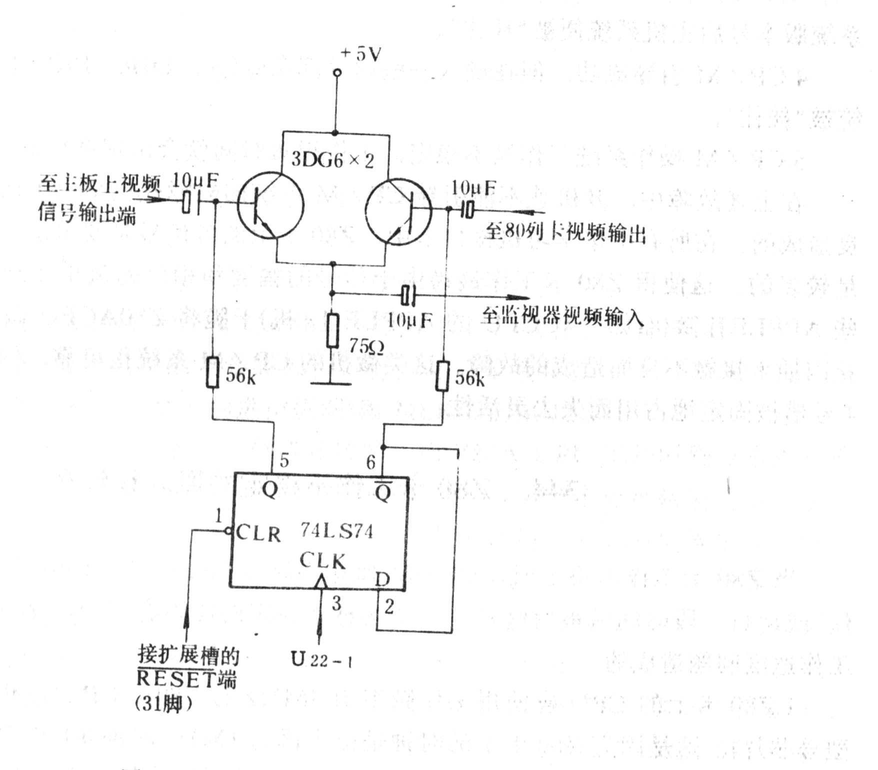


图 342-2 程控切换显示方式接线图

使用时只要开机则切换电路就选择了 80 列文本显示方式。但是此时若载入的不是 CP／M 操作系统或 PASCAL 操作系统而是 DOS3.3，则还必须键入“PR#4”(设 80 列卡插在第 4 号槽)才能有 80 列文本显示。以后，若需要显示高分辨率图形时，只要触动地址 $ C040 (DOS3.3 或 BASIC 状态时)一次。若使用 CP／M 系统则要触动 E040H 地址。一次。若再要回到 80 列文本时，就再角动一次 $ C040。这一命令在监控下可用“LDA $ C040”、“BIT $ C040”或“STA $ C040”等，在 BASIC 状态下则可用“POKE−16320,0”或“PEEK(−16320)”命令来触发。这些命令可编排在程序中，在程序运行时自动地对显示方式进行切换。

343. Z80 卡常见故障现象有哪些

中华学习机插上 Z80 CPU 卡后，便可以使用 CP／M 操作系统。该系统用途广泛、软件丰富、便于改造或移植，是一种非常流行的操作系统。但是如果 Z80 卡有故障、性能不好或与主机配合不好，则 CP／M 操作系统便不能正常工作。常见 Z80 卡故障现象如下所述：

1.插上 Z80 卡后不能引导 CP／M 系统、也不能引导 DOS3.3 系统，有时甚至主机不

能在开机时发出“嘟”的一声响。

2.可以正常引导 DOS3.3，但不能引导 CP／M 系统。插入 CP／M 系统磁盘并开机后，Z80 卡上的“CARD ON”指示发光二极管不亮。

3.开机后可以载入 CP／M 操作系统，但屏幕显示杂乱字符或只显示一部分 CP／M 系统版本号后主机系统便被“挂住”。

4.CP／M 引导成功，但在键入一些内外部命令(如：DIR、PIP、MBASIC 等)时，系统被“挂住”。

5.CP／M 操作系统工作极不稳定，工作很短时间就会出现系统被“挂住”的现象。

在上述故障中，开机就不能引导 CP／M 系统的故障往往是由于插卡不到位或接触不良造成的。在所有中华学习机接口卡中，Z80 卡上的各信号是要求最严格而信号的合成又是较多的。这使得 Z80 卡工作最易由于信号的幅度和相位的微小差异而遭破坏。所以有些 APPLEⅡ微机(如：双 CPU 的 APPLEⅡ+机)干脆将 Z80ACPU 做到了主板上，以防止因插卡接触不良而造成的故障。这类微机的 CP／M 系统很可靠，但是付出的代价是第 4 号槽被固定地占用而失去灵活性。

344. Z80 卡工作不稳定的原因是什么

当 Z80 卡工作不稳定时，CP／M 操作系统虽然被引导，但引导之后主机系统被“挂住”或运行一段时间后被“挂住”。产生这种不稳定的原因有三个，基本上都是电路元器件工作速度问题造成的。

1.Z80 卡上的 CPU 应使用工作频率为 4MHz 的 Z80A CPU(或相同功能的其他系列型号芯片)。这是因为 Z80 卡上的时钟是由主机的 1MHz 时钟 Φ1 和 7MHz 时钟合成后再分频而成的，它是一个平均频率为 2MHz 的非对称时钟信号。它有 1 个半的 3.58MHz 周期和半个 1MHz 周期（如图 344-1 所示）。所以在工作时，要求 CPU 能有 4MHz 的反应能力。但是，由于一些 Z80 卡上使用了 2MHz 工作频率的 Z80 CPU 或使用了一些频率特性不够好的 CPU，使得系统工作一段时间或还未能工作，就因反应不过来而使系统死掉。

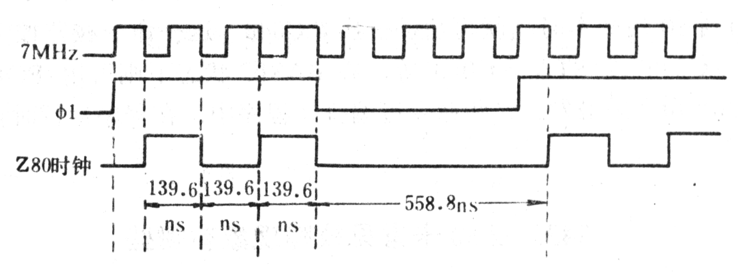


图 344-1　Z80 卡上的主时钟信号

2. Z80 卡时钟产生电路应使用工作速度较高的 74S20 芯片，并且为加速和改善波形，电路里还采用了由晶体管 Q1(2N3906)及一些阻容元件构成的加速电路。如果 Z80 卡上选用了工作速度较低的 74LS20 芯片来代替 74LS20，或晶体管 Q1 及阻容元件性能变差，都

会导至卡上 Z80A CPU 时钟波形变差，而使 CPU 不能正常工作。

3.时钟分频电路使用的双 JK 触发器(74LS107)对触发脉冲的要求较为严格。当由于 7MHz 信号整形电路中 R10、R11、R12、C8 各元件参数不准或变值时，都会影响 JK 触发器的触发信号的占空比，而使 74LS107 不能正常工作。所以 CPU 也就不能正常工作。此时可选用高速的 74S107 或 74F107。

此外，主机电源的干扰和主板总线驱动电路的工作速度也会对 Z80 卡的正常工作造成影响，同样不应忽视。

345. Z80 卡上的四个小开关是干什么用的

在 Z80 卡上有四个小开关。这四个小开关一般情况下总是置于"OFF"(关)的位置。如果你做了一个实验,就会发现:这四个开关中除了开关 1 以外，其它三个开关无论放置什么状态都不影响 CP／M 操作系统的正常载入和运行。开关 1 接在芯片 74LS283(四位二进制全加器)的第 7 脚上(该脚为全加器的进位位端)和 74LS86 的第 12 脚。

6502 CPU 工作时其内存地址是被分成几段的。而 Z80A CPU 工作时则要求内存地址是连续的。为了解决这一地址上的矛盾，在 Z80 卡上安装了一个地址映射电路，它由 74LS283.74LS138 和 74LS86 以及 74S20 的一半构成。它的作用是将 Z80 的地址 0000H～AFFFH 和 F000H～FFFFH 分别加 1000H，成为 6502 CPU 的 $1000～$BFFF 和 $0000～$0FFF 地址段；将 Z80 CPU 的地址 B000H～DFFFH 分别加 2000H，成为 6502 CPU 的 $D000～$EFFF 地址段；而将 Z80 CPU 的地址 E000H～EFFFH 直接转化为 6502 CPU 的 $C000～$CFFF 地址段。这样处理后 Z80 CPU 工作时躲过了 6502 CPU 的系统 RAM $0000～$07FF。使 Z80 CPU 的 B000H～DFFFH 和系统 RAM 连在一起放到了 Z80 CPU 地址的最高位 E000H～FFFFH，这使得 Z80 CPU 有了一个连续地址的内存分配。这种地址的映射在 Z80 卡上的开关 1 断开时("OFF")是可由电路实现的。而当开关 1 闭合时("ON") Z80 CPU 地址仅被缓冲，而不经变换(映射)地出现在中华学习机的地址总线上。如果此时引导的 CP／M 系统是处理映射后的地址的操作系统，则就会因地址不对而出错。

开关 2.3.4 分别是控制 Z80 CPU 的 BUSRQ(总线请求，即 DMA 请求)、NMI(非屏蔽中断)、IRQ(可屏蔽中断)端的。它们的意义如下：

开关	开("OFF")	关("ON")
S1-1	地址映射	地址仅缓冲、不映射
S1-2	允许 Z80 DMA	Z80 DMA 断开
S1-3	允许 Z80 NMI	Z80 NMI 断开
S1-4	允许 Z80 IRQ	Z80 IRQ 断开

346. 有些中华学习机不能使用 Z80 卡是什么原因

有些中华学习机的故障只是在主机插上 Z80 卡载入 CP／M 操作系统时才出现。此时，系统不能正常载入；主机被“挂起”；屏幕上是一些杂乱字符；按“CTRL-RESET”键

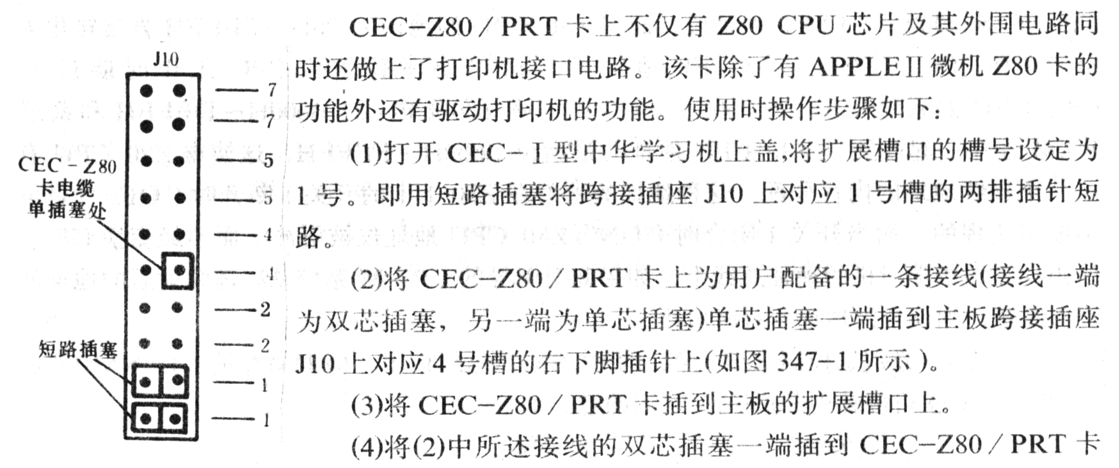
亦无效。

这些中华学习机在DOS系统、Pascal操作系统、LOGO等系统下工作都正常。主机自检也不见有任何不正常现象。Z80卡完好，且在其它中华学习机上可以使用。主机扩展插口上各信号均无异常。

Z80卡上的Z80A CPU与主机6502 CPU在CP／M操作系统下是交替工作的。当Z80A CPU使用主机RAM时，卡上要产生DMA信号和RDY信号。其中DMA信号(直接存储器存取信号)要中断主机6502 CPU的时钟Φ0，并使6502 CPU地址总线驱动器电路（U2、U3）呈高阻态而让出地址总线。RDY信号接到6502 CPU的第2脚(RDY端)。当RDY信号变低且6502 CPU在读动作时，该信号有效。它将使6502 CPU产生一个延长的读周期，而不对数据总线有驱动。于是Z80卡上的Z80A CPU便可占用数据总线。所以，当主机板上的U36芯片（74LS02）损坏时，将使DMA信号不起作用。而当主机6502 CPU的RDY端内部损坏时（笔者曾见两例该故障)，将使RDY信号不起作用。这些故障都是造成CP／M操作系统不能正常工作的原因。

347. 怎样使用中华学习机CEC-Z80／PRT卡

CEC-Z80／PRT卡是专为CEC-Ⅰ型中华学习机设计的Z80卡。插上该卡后，可使用CP／M操作系统，运行CP／M操作系统下的BASIC程序、FORTRAN程序和d-BASEⅡ数据库管理程序等。



CEC-Z80／PRT卡上不仅有Z80 CPU芯片及其外围电路同时还做上了打印机接口电路。该卡除了有APPLEⅡ微机Z80卡的功能外还有驱动打印机的功能。使用时操作步骤如下：

(1)打开CEC-Ⅰ型中华学习机上盖,将扩展槽口的槽号设定为1号。即用短路插塞将跨接插座J10上对应1号槽的两排插针短路。

(2)将CEC-Z80／PRT卡上为用户配备的一条接线(接线一端为双芯插塞，另一端为单芯插塞)单芯插塞一端插到主板跨接插座J10上对应4号槽的右下脚插针上(如图347-1所示)。

(3)将CEC-Z80／PRT卡插到主板的扩展槽口上。

(4)将(2)中所述接线的双芯插塞一端插到CEC-Z80／PRT卡左上角的两个插针上。以后便可正常用了。

图347-1 CEC-Z80／PRT卡接线图

(5)启动CP／M操作系统磁盘，卡上的发光二极管应变亮,表示工作正常。

(6)若在不用该卡而想将其拔下时，可先将(2)中所述接线双芯插塞一端拔下，然后再将该卡拔出。连线另一端不动，以勉下次再用时还需打开主机上盖。

(7)若只想将该卡当打印卡使,可参照本书第348题中所述方法进行操作。

348. 怎样使用中华学习机汉字打印卡

中华学习机汉字打印卡是在原APPLEⅡ微机打印机接口卡电路的基础上进行逻辑化

简的结果。将 APPLEⅡ微机打印机接口卡上的 11 块芯片(包括 ROM)化简为仅用两块芯片代替的汉字打印卡。该卡保留了原卡的基本打印功能，而且在汉字方式下可以完全代替原卡。该卡的显著优点在于：

1.价格下降为原卡的 1／5−1／8。

2.由于体积小，可方便地从 CEC−I 型中华学习机上盖的小门处将该卡插到扩展插槽上。

3.该卡在西文方式下工作时使打印速度加快。

中华学习机汉字打印卡的使用方法如下：

1.将该卡插在 CEC−I 型中华学习机的扩展插槽上，并将该槽设定为 1 号槽。若为 APPLEⅡ微机，则应将卡插在 1 号槽上。

2.用专用平行电缆将该卡与打印机接口连接好。

3.打开主机及打印机电源开关。使中华学习机进入汉字方式。若为 APPLEⅡ微机，则可启动硬汉卡或载入软磁盘汉字系统。

4.使用汉字方式下常规的各种打印命令进行打印工作。

5.若希望在中华学习机或 APPLEⅡ微机的西文方式下打印，则可使用配合该打印卡的 DOS 磁盘(经过小小改动的 DOS3.3)。而且该磁盘上还提供一个可使用“CTRT−Q”命令打印高分辨率图形的辅助文件。

6.若希望在西文方式下打印，且主机又无磁盘驱动器，则可先键入下面的一小段程序(在监控状态下进行)：

```
0300-  48  20  F0  FD  2C  00  C1  30
0308-  FB  68  29  7F  8D  90  C0  C9
0310-  0D  D0  05  A9  0A  48  D0  EC
0318-  60   A9  00  85  36  A9  03  85
0320-  37  60  A9  F0  85  36  A9  FD
0328-  85  37  60
```

然后回到 BASIC 状态。需要打印机时可用命令“ CALL 793 ”联打印机。打印完毕后可用命令“ CALL 802 ”脱开打印机(亦可用 CTRL−RESET 键)。

7.在 CP／M 操作系统下，接通和使用打印机的一切操作均不变。

该打印卡特别适用于学校微机室中多台主机共用一台打印机的情况。此时，可在不关机、不拔卡的情况下分别连接使用打印机。

349. 怎样判断打印卡故障

打印卡电路损坏而造成的打印故障是较为常见的。它主要发生在打印驱动程序的固化 ROM 电路，STROBE 信号产生电路，ACKLNG.BUSY 信号接收电路和数据输出驱动电路。以下分别叙述各故障现象及检修方法。

1.当打印卡上的 2716 EPPOM 损坏时，内部固化的打印驱动程序被破坏。在 BASIC 状态下键入“PR#1”(设打印卡插在 1 号槽)，则屏幕上出现杂乱字符，且系统被“挂住”。要想确定仅是 2716 EPROM 损坏而其它电路正常，可运行下面的程序：

```
         ORG   $300
START:   LDA   $C1C1          0300-AD  C1  C1
         BMI   START          0303-30  FB
         LDA   #$41           0305-A9  41
         STA   $C090          0307-8D  90  C0
         JMP   START          030A-4C  00  03
```

当打开打印机，在监控状态下敲入上述程序并键入"300G"后，打印机连续打印字符"A"说明除 2716 EPROM 外其它电路无故障。应更换 2716 EPROM。

2.当 STROBE 信号产生电路故障时，主机系统在敲入" PR#1 "后将被“挂住”。这时，可用逻辑笔检测打印卡上的 IC 芯片 74LS74 的第 8 脚。若该点不是低电平，则应更换 74LS74 后再试一试。如果 74LS74 脚正常，应再检测两片相同 IC 芯片 74LS07 或(74LS17)上面一片的第 12 脚是否为低电平。若不为低，则 74LS07 损坏。一般该电路损坏的情况较多。

3.当 ACKLNG 和 BUSY 信号接收电路故障时，现象与 2 相同。这时可用逻辑笔检测打印卡上的 74LS14 的第 3 脚是否为低电平，若不为低，则是打印机故障或信号线连接故障。若为低，应再查 74LS14 的第 4 脚是否为高，74LS00 的第 3 脚是否为低。若不正常，可直接发现故障点。若正常，则说明 74LS251 可能损坏。

4.当数据输出驱动电路故障时，往往表现于打印错误字符，而且这些错误都具有一定规律。通过在打印时检查 8 条数据线的固"1"或固"0"的信号线，便可查出故障的 IC 芯片。有时，是驱动电路 74LS07 损坏，有时是三态锁存器 74LS373 损坏所至。

当然在出现故障时也不能完全排除其它 IC 电路损坏的可能性。但由于其它 IC 芯片与外设信号无直接接触，损坏的概率比较小。

350. 怎样自制接口插座扩展箱

中华学习机只给用户提供一个扩展插槽，当用户希望同时接两个扩展卡时将无能为力。这里给读者介绍一种自制扩展箱的方法。中华学习机配上该扩展箱后可同时插两、三个扩展卡。并且一些原插不进中华学习机的 APPLE Ⅱ 微机插卡也可使用了。制作方法如下：

(1) 买两、三个标准的 50 线插座(APPLE Ⅱ 总线标准)。再找一个(或制一个) 50 线的插脚板。

(2) 将 50 线插座并排(之间相隔 3～5 厘米)固定在一块塑料板上(见图 350-1)。

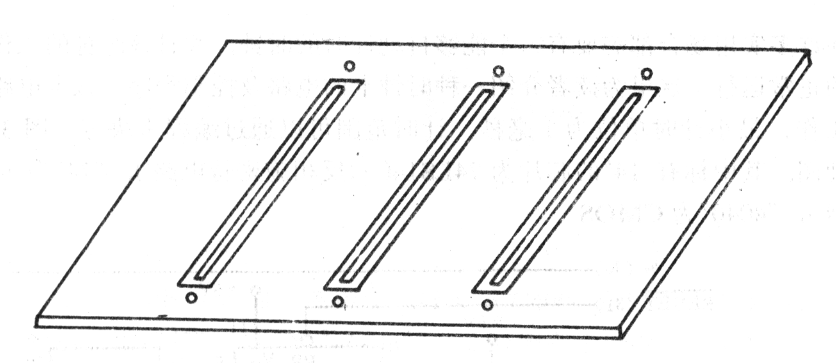


图 350−1　扩展箱插座的安装

(3) 除第 1、23、24、27、28、41 脚之外，将几个 50 线插座的其余 44 脚分别连接起来，并与插脚板上各对应脚相连。

(4) 主板上跳接插座 J10 对应每个槽号的 4 个插针中，右下插针为 I／O SEL 信号，右上插针为 DEV SEL 信号。请将它们分别引到自制扩展箱 50 线插座的第 1 脚(I／O SEL)和第 41 脚(DEV SEL)。各插座的对应槽号应以选择第 1 号、第 2 号、第 4 号比较好。其中第 1 号常用于插打印机卡，第 2 号槽常用于插串行通讯卡或 A／D 、D／A 卡，第 4 号槽常用于插 Z80 CPU 卡。

该扩展箱安装好后将 50 线插脚板插在中华学习机主板的扩展槽上即可。

351.　主机电源的常见故障有哪些

中华学习机主机电源产生故障原因是多方面的。如：过长时间的开机不关，而周围环境温度又太高，将造成电源输入端的损坏。带电插拔接口卡或磁盘驱动器电缆插反，可造成电源输出端的损坏。

电源输入端损坏时，无输出电压，电源指示发光二极管不亮。一般损坏的元件是全波桥式整流二极管及高压开关管。

电源输出端有四种电压输出，分别是+5V,+12V,−5V,−12V。带电插拔接口卡可能造成+5V 或+12V 短路，使该电路的整流二极管烧毁。磁盘驱动器电缆插反，使 +12V 与 −12V 位置对调，若驱动器磁头读出放大电路芯片不损坏，则会使 −12V 的整流二极管烧毁。烧毁后的二极管一般呈短路状态，可用万用表的低阻档在电源板上直接测量。+5V 与 +12V 整流二极管损坏时，电源指示发光二极管也不亮。但−5V 或−12V 电源故障时，在开机或关机瞬间电源指示发光二极管闪耀一下。

中华学习机主机电源除上述常见故障外，还有一些不常见的故障。如:电源输出电压过高或不可调，电源负载能力不够而“达达”响，电源打火而影响主机的正常工作等。这些故障分别可能是变压器耦合电路坏，元器件质量差和接触不良造成。但因这些是非常见故障，所以不再详述。

352. 怎样自制一个时钟卡

在许多实用场合都需要有一个能够自动计时的时钟，并且该时钟的工作不应影响用户程序的正常运行。这里为读者介绍一种时钟卡的电路及控制程序。该卡电路以非屏蔽中断方式工作，最小计时单位为 1 毫秒，计时范围可以通过编程来决定。图 352−1 为该卡的电原理图。其中标有″14″的芯片为 74LS14(六反相施密特电路)，″74″为 74LS74(双 D 型触发器)，″4040″为 CMOS

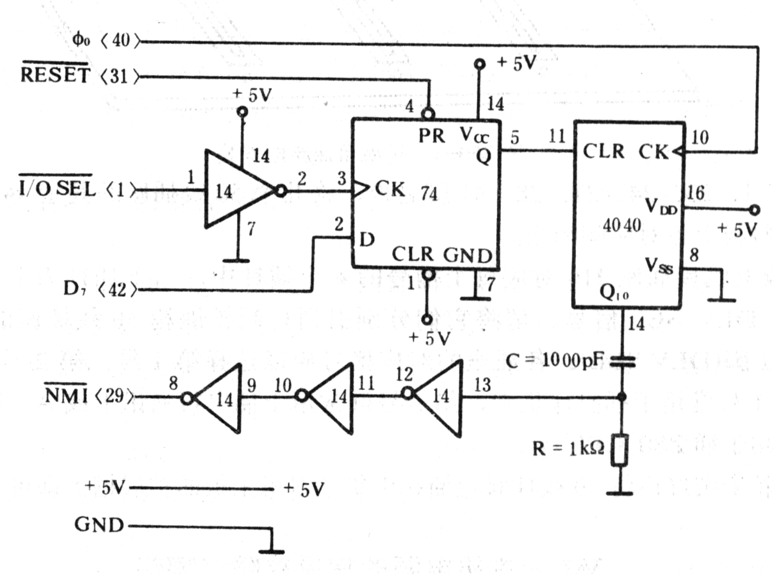


图 352−1 时钟卡电路原理图

的 12 位计数器 4040。电阻 R 和电容 C 构成微分电路，以产生中断请求的窄脉冲信号，74LS14 用于对该微分信号整形。74LS74 用于控制 4040 的动作。而 4040 对系统时钟 $\varphi 0$ 分频后大约每一毫秒产生一次中断请求。该卡的控制程序如下：

(设该卡被设定为 4 号槽)

```
0300- 48           PHA
0301- A9 10        LDA  #$10
0303- 8D FC 03     STA  $03FC
0306- A9 03        LDA  #$03
0308- 8D FD 03     STA  $03FD
030B- 8D 00 C4     STA  $C400
030E- 68           PLA
030F- 60           RTS
0310- E6 06        INC  $06
0312- D0 03        BNE  $0316
0314- E6 07        INC  $07
0316- 40           RTI
```

计时以毫秒为单位存储在零页 \$06 和 \$07 单元中。\$06 为低 8 位，\$07 为高 8 位。计时范围为 0～65535 毫秒。若需要增加计时范围可改写这小段程序：

```
      .
      .
      .
0312- D0  07           BNE   $031A
0314- E6  07           INC   $07
0316- D0  03           BNE   $031A
0318- E6  08           INC   $08
031A- 40               RTI
```

则计时可达0～16777215毫秒。

该程序中从地址 $030B 开始的一条指令“STA $C400”是假设该卡被设定为4号槽。若设定为1号槽，则用指令“STA $C100”；2号槽则用“STA $C200”，以此类推。

353. 怎样灵活使用128K扩展RAM卡

生产128K卡的厂家为用户配置了丰富的系统软件，如:128K DOS、128K CP／M和128K Pascal等。但若读者希望更加灵活地运用128K卡，使其为自已的一些特殊目的服务(例如：在考贝磁盘时将原磁盘数据全部读入128K卡内)，则需要对128K卡的操作地址有所了解才行。在表353-1中列出了128K激活及关闭的软开关地址。

表353-1 128K卡软开关地址

地址	功能
$Con0	选择4K体A RAM可读不可写
$Con1	选择4K体A ROM可读、RAM可写
$Con2	选择4K体A ROM可读、RAM禁读写
$Con3	选择4K体A RAM可读可写
$Con4	选择16K体1
$Con5	选择16K体2
$Con6	选择16K体3
$Con7	选择16K体4
$Con8	选择4K体B RAM可读不可写
$Con9	选择4K体B RAM可读、RAM可写
$ConA	选择4K体B ROM可读、RAM禁读写
$ConB	选择4K体B RAM可读可写
$ConC	选择16K体5
$ConD	选择16K体6
$ConE	选择16K体7
$ConF	选择16K体8

表中地址的“n”为 128K 卡所在槽号。关于 16K 体和 4K 体的概念请见图 353–1。

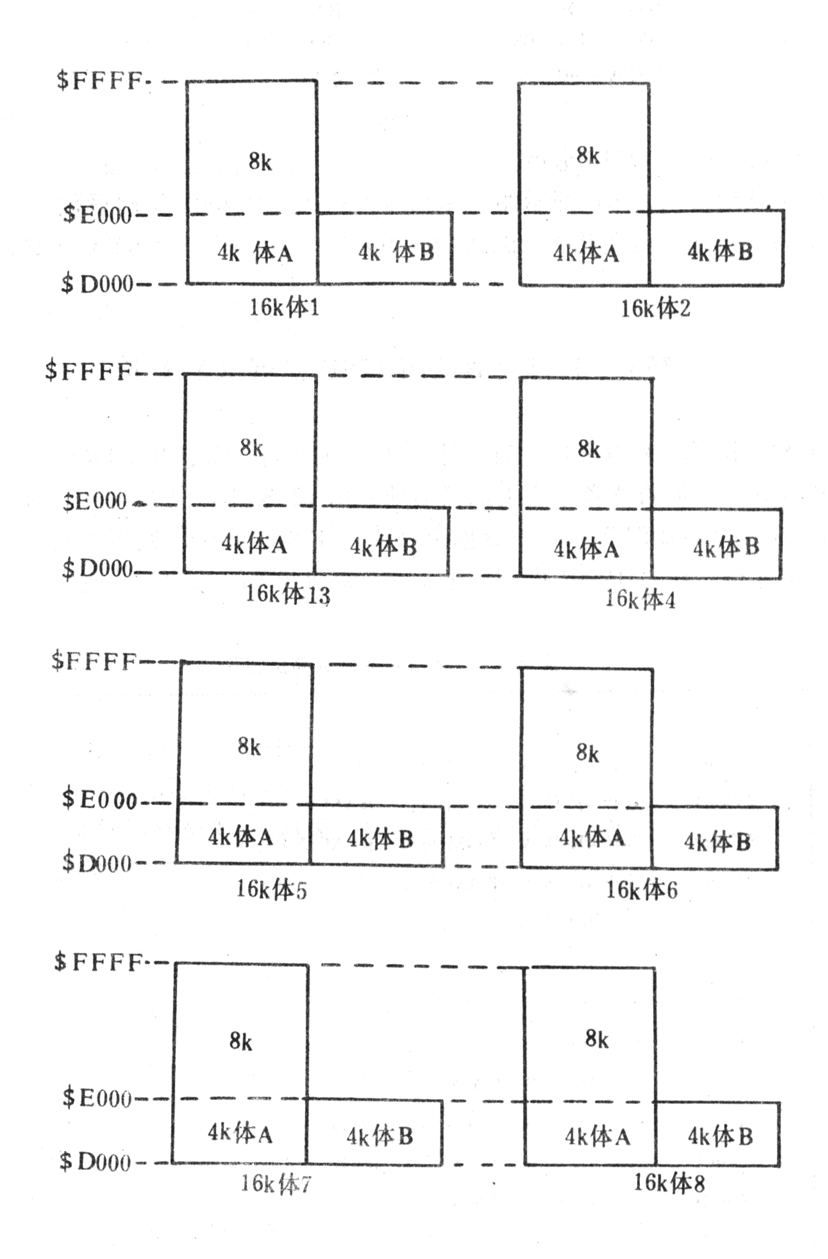


图 353–1　128K RAM 空间分配图

354. 怎样将中华学习机中的数据传到 TP801 单板机中

用中华学习机在 CP／M 操作系统下的 Z80 宏汇编编出的程序，一般都是存在磁盘上。若需要到 TP801 单板机上实际运行这些程序时，就要将程序的机器码打印出来，再键入到单板机中。这样做十分麻烦。若程序过长，还容易出错。这里给读者介绍一个将中华学习机中的 Z80 机器语言程序传到 TP801 单板机中的方法。

传送是通过中华学习机上打印卡进行的。首先将 TP801A 单板机上 Z80PIO 的第 18 脚（ARDY）与 U26（74LS04）的第 3 脚用导线连接起来，再将 U26 的第 4 脚与中华学习机打印卡 20 芯电缆的第 20 线（BUSY）连接。其余各脚按表 354–1 连线。

表 354–1 打印卡与 PIO 间的连接

中华学习机打印卡 20 芯插座	Z80PIO 管脚号
1 脚（STROBE）	16 脚（ASTB）
3 脚（D0）	15 脚（PA0）
5 脚（D1）	14 脚（PA1）
7 脚（D2）	13 脚（PA2）
9 脚（D3）	12 脚（PA3）
11 脚（D4）	10 脚（PA4）
13 脚（D5）	9 脚（PA5）
15 脚（D6）	8 脚（PA6）
17 脚（D7）	7 脚（PA7）
2 脚（GND）	11 脚（GND）

将下列 Z80 机器语言程序键入 TP801A 单板机从 2C00H 开始的 RAM 区中：

```
3E  2C  ED  47  3E  1B  D3  82  3E
4F  D3  82  3E  87  D3  82  21  00
20  ED  5E  FB  DB  00  76  18  FD
1D  2C  DB  80  77  23  FB  ED  4D
```

启动中华学习机的 CP／M 操作系统，并用“DEBUG”将要传送的程序载入中华学习机内存。然后进入 6502 监控（请参见本书 CP／M 操作系统有关题目）。再将下列 6502 机器语言程序键入从 $ 300 开始的 RAM 区中：

```
0300-   A0 00       LDY  #$00
0302-   A9 00       LDA  #$00
0304-   85 06       STA  $06
0306-   A9 11       LDA  #$11
0308-   85 07       STA  $07
030A-   B1 06       LDA  ($06), Y
030C-   2C C1 C1    BIT  $C1C1
030F-   30 FB       BMI  $030C
0311-   8D 90 C0    STA  $C090
0314-   C8          INY
0315-   D0 F3       BNE  $030A
0317-   60          RTS
```

然后使 TP801 单板机从 2C00H 开始运行，使中华学习机从 $ 300 开始运行（用“＊300G”命令）。则可将中华学习机从 $ 1100 单元开始的待传送机器码传到 TP801 单板

机从 2000H 单元开始 RAM 中。

待传送完毕后，中华学习机屏幕上光标再次出现。此时按 TP801 单板机的“MON”键，使其从接收程序中退出。

此程序最多可传送 256 字节的机器码。若读者希望传送更多数据，可对 6502 机器语言程序做一些改动。

355. 你知道“万能接口卡”吗

“万能接口卡”是一种配合 APPLEⅡ微机及其兼容机（中华学习机），主要用于学习和开发 MCS−51 系列单片微机的多功能智能卡。但在不是用于单片微机的学习和开发时，插在 APLLEⅡ微机上可以做为一个串行通讯接口卡（可用于 APPLEⅡ微机与 IBM PC 等微机之间通讯、联网），或一个打印机接口卡（可用于全脱机方式打印），或一个时钟卡（可显示时间、控制主机定时动作），或并行接口卡，或 PC 磁盘驱动器兼容卡…。只要配上相应的软件，就可方便地用于再开发，灵活地完成各种功能。

“万能接口卡”所以万能，是由于该卡上有一个 MCS−51 系列单片微机 8031AH。MCS−51 系列单片微机是目前国内应用最广泛、功能极强、价格低廉、性能价格比最优的一类芯片。“万能接口卡”上的 8031 单片微机与主机 6502CPU 并行工作。8031 单片微机可以寻址主机的所有存储器空间和 I／O 空间，完成诸如：控制、显示、发声、打印、键盘输入、管理 DOS 等功能，以至接管主机整个系统，使主机成为一台以 8031 为 CPU 的“APPLEⅡ”微机。

该卡在用于单片微机的学习和开发时有下列明显优点：

(1) 允许用户使用零页程序存储区，且逻辑地址与物理地址完全一致，所以对用户的透明度极高。用户用于开发时对用户目标机无地址译码的限制。这是一般单片微机开发系统所不能相比的。

(2) 在调试用户程序时，除了单步跟踪、断点、连续运行方式外，还具有单拍运行方式（一拍为 8031 的一个机器周期）。这使对硬件开发感兴趣的用户在进行目标机调试时感到十分方便。这也是一般单片微机开发系统所不具备的。

(3) 该卡用于单片微机的学习、开发时，所配置的支持软件除了一般的编辑(EDIT)、汇编（ASM51)、调试纠错（DEBUG）程序外，在 DEBUG 中还增加了单拍调试、查找、块移动和对单片微机在片 RAM（包括可寻址的位）及所有寄存器进行显示和修改的功能。

(4) 为了更好地体现单片机学习器的功能，在系统软件中配有一些通用子程序。8031 单片微机可以通过系统调用来实现对整个主机系统的管理。

(5) 该卡在设计构思上是一个独创，所以功能很强，而价格却很低廉（售价仅 300 元左右)，如此高的性能价格比是同类产品不及的。

356. 如何用“万能接口卡”开发 8031 单片微机用户系统

使用“万能卡”（AP51−I 型卡）可开发单片用户机（8031)。操作方法如下：

(1) 将 AP51–I 型卡插在中华学习机扩展插槽（或 APPLE Ⅱ微机任意扩展槽）上。扩展槽号可任意设定。

(2) 启动配合该卡的“MCS–51 学习开发系统”磁盘。屏幕将显示：

————————AP51–I————————

MCS–51 LEARNING & ESPLOITING SYSTEM

COPYRIGHT AI LUN & DONG LE 1989 VER1.0

ENTER A KEY TO CONTINUE. ? FOR HELP

此后，若按“?”键则显示各种命令及其功能的菜单。按其它键则进入系统，并显示提示符“–”。

(3) 键入表 356–1 所示的命令，可实现对 8031 单片机系统的各种操作。命令主体一般由一两个字母构成，参数部分“add1”表示十六进制开始地址，“add2”表示十六进制结束地址，字母 n 则为十六进制整数。

表 356–1 AP51–I 命令一览表

命令格式	命令功能
DR↓	显示 8031 内各寄存器的内容
DD add1，add2↓	显示 8031 在片 RAM 的内容
DC add1，add2↓	显示用户程序区和数据区 RAM 的内容
DB add1，add2↓	显示 8031 内部可寻址位的内容
SD add1，add2↓	显示并修改 8031 内在片 RAM 的内容
SC add1，add2↓	显示并修改用户程序区和数据区 RAM 的内容
SB add1，add2↓	显示并修改 8031 内部可寻址的位
键入各寄存器名↓	显示并修改指定寄存器的内容
AS add1↓	从 add1 开始对键入的助记符进行汇编
U add1↓	从 add1 开始进行反汇编
S add1↓	从 add1 开始单步运行用户程序
T n↓	退踪运行 n 条指令
G add1↓	从 add1 开始全速运行用户程序
Q ↓	退出系统到 BASIC 状态
.↓	从修改、汇编状态退出
SV 文件名 add1,add2	将 add1 至 add2 的用户程序以指定文件名存到磁盘上
LD 文件名 add1	将磁盘上指定名文件载入从 add1 开始的用户区。

(4) 将该卡上的仿真头插在用户系统的单片机位置，可对用户机进行在线仿真。

357. 怎样实现“万能接口卡”的时钟卡功能

“万能卡”（AP51–I 型卡）具有时钟卡的功能，这只须将该卡插在中华学习机扩展槽上（设定 1 号槽），然后运行一个相应的程序即可。这个程序及键入方法如下：

(1) 将表 357–1 所列程序清单的左边第二列数据键入中华学习机从 $ 300 开始的存储器中（共 $ 80＝128 个单元）。然后以 TIME 为文件名存到磁盘上。

表 357-1　TIME　程序清单

```
                              ORG 0300H
0300 E4          START:CLR    A
0301 F530              MOV    30H,A
0303 F531              MOV    31H,A
0305 F532              MOV    32H,A
0307 F533              MOV    33H,A
0309 F589              MOV    TMOD,A
030B D28C              SETB   TRO
030D D2AF              SETB   EA
030F D2A9              SETB   ETO
0311 E533        LP1:  MOV    A,33H
0313 B40BFB            CJNE   A,#OBH,LP1
0316 E4                CLR    A
0317 F533              MOV    33H,A
0319 E532              MOV    A,32H
031B 2401              ADD    A,#1
031D D4                DA     A
031E F532              MOV    32H,A
0320 B46017            CJNE   A,#60H,DISP
0323 E4                CLR    A
0324 F532              MOV    32H,A
0326 E531              MOV    A,31H
0328 2401              ADD    A,#1
032A D4                DA     A
032B F531              MOV    31H,A
032D B4600A            CJNE   A,#60H,DISP
0330 E4                CLR    A
0331 F531              MOV    31H,A
0333 E530              MOV    A,30H
0335 2401              ADD    A,#1
0337 D4                DA     A
0338 F530              MOV    30H,A
033A E530        DISP: MOV    A,30H
033C 7170              ACALL  CH
033E 90E420            MOV    DPTR,#0E420H
0341 E535              MOV    A,35H
0343 F0                MOVX   @DPTR,A
0344 E534              MOV    A,34H
0346 A3                INC    DPTR
0347 F0                MOVX   @DPTR,A
0348 E531              MOV    A,31H
034A 7170              ACALL  CH
034C 90E423            MOV    DPTR,#0E423H
034F E535              MOV    A,35H
```

```
0352 A3            INC     DPTR
0353 E534          MOV     A,34H
0355 F0            MOVX    @DPTR,A
0356 E532          MOV     A,32H
0358 7170          ACALL   CH
035A 90E426        MOV     DPTR,#0E426H
035D E535          MOV     A,35H
035F F0            MOVX    @DPTR,A
0360 A3            INC     DPTR
0361 E534          MOV     A,34H
0363 F0            MOVX    @DPTR,A
0364 74BA          MOV     A,#0BAH
0366 90E422        MOV     DPTR,#0E422H
0369 F0            MOVX    @DPTR,A
036A A3            INC     DPTR
036B A3            INC     DPTR
036C A3            INC     DPTR
036D F0            MOVX    @DPTR,A
036E 6111          AJMP    LP1
0370 F8        CH: MOV     RO,A
0371 540F          ANL     A,#0FH
0373 44B0          ORL     A,#OBOH
0375 F534          MOV     34H,A
0377 E8            MOV     A,RO
0378 C4            SWAP    A
0379 540F          ANL     A,#OFH
037B 4BO           ORL     A,#OBOH
037D F535          MOV     35H,A
037F 22            RET     35H,A
        INCLUDE IN HASHTAB:
         START:      300
           LP1:      311
          DISP:      33A
            CH:      370
```

(2) 将表 357-2 所列程序清单中左边第二列数据键入中华学习机从 $ 2000 单元开始的内存储器中（共 15 个单元）。并以 TIME1 的文件名存到磁盘上。

表 357-2 TIME1 程序清单

```
                                    ORG    2000H
2000 02E300              LJMP   0E300H
2003 00                  NOP
2004 00                  NOP
2005 00                  NOP
2006 00                  NOP
2007 00                  NOP
2008 00                  NOP
2009 00                  NOP
200A 00                  NOP
200B 00                  NOP
200C 0533                INC    33H
200E 32                  RETI
        INCLUDE IN HASHTAB:
```

(3) 键入下列 BASIC 程序，并以 TIMER 为名存到磁盘上。

```
10 D$=CHR$(4):PRINT D$;"BLOAD TIME,A$300,L$90"
20 PRINT D$;"BLOAD TIME1,A$2000,L$10"
30 POKE 49412,0
40 FOR I=0 TO 1000:NEXT
50 POKE 49414,0
60 HOME:NEW
```

(4) 开机载入 DOS 后，只要键入"RUN TIMER ↓"则不久就会在屏幕右上角出现显示的时间。

358. 怎样使"万能接口卡""奏乐"

上一题目介绍的"万能接口卡"（AP51-I）具有一个独特的功能，它能在您正常使用中华学习机的同时（除使用磁盘外），让中华学习机为您奏响美妙的音乐。这可使您边操作中华学习机（如键入、修改 BASIC 程序，运行程序等），边听着乐曲。方法如下：

(1)、将"万能"卡插在中华学习机扩展插槽上，并设定为 1 号槽。

(2)、进入监控状态，并从 $2000 单元开始键入表 358-1 所列 8031 机器码。

表 358-1 8031 单片机程序机器码

```
2000-90 00 30 7E 2A E0 FD A3
2008-E0 A3 CD 11 11 DE F6 01
2010-00 C0 82 C0 83 90 A0 30
2018-7C 01 F0 DC 07 DD 05 D0
2020-83 D0 82 22 DB F5 FB 01
2028-1A 8D 94 C0 8D 96 C0 60
2030-2B 11 30 28 30 07 2B 07
2038-26 1B 20 07 26 11 39 11
2040-2B 11 30 28 30 07 2B 07
2048-26 11 30 07 2B 07 26 11
2050-39 11 26 07 20 07 1C 28
2058-18 11 1C 28 20 11 26 11
2060-1C 11 26 28 2B 07 26 07
2068-30 28 30 07 2B 07 26 07
```

2070–20 07 26 28 2B 11 30 11
2078–39 11 30 28 39 07 30 07
2080–39 28 39 28 25 2A 32 28
2088–3C 40 2B 00 00 00 00 00

(3)在监控状态下键入下列命令:

* C104:00↓
* C106:00↓
* (CTRL–C)↓

于是中华学习机将奏出音乐，而毫不影响对它进行各种操作。按″CTRL–RESET″键可使音乐停止，键入(3)中各命令又开始奏乐。

359. 怎样让中华学习机“说话”

CEC–Ⅰ中华学习机的喇叭不仅可以放出音乐，还可以放出预先录制好的说话声。其方法是利用一段程序将采集的数据变成触动喇叭发声电路的动作。具体操作步骤如下:

(1)将表 359–1 所列机器码键入从 ＄300 开始的单元。然后可将该段程序机器码存入磁盘上。

表 359–1 发声程序机器码

```
0300–A9 00 8D 00 10 EE 03 03
0308–D0 F8 EE 04 03 AD 04 03
0310–C9 96 D0 EC A9 00 8D 03
0318–03 A9 10 8D 04 03 A0 05
0320–88 D0 FD AD 60 C0 29 80
0328–CD 58 03 8D 58 03 D0 05
0330–E8 E0 7F D0 E9 8A A2 00
0338–4D 58 03 8D 00 10 EE 3C
0340–03 D0 DB EE 3D 03 AD 3D
0348–03 C9 96 D0 D1 A9 00 8D
0350–3C 03 A9 10 8D 3D 03 60
0360–AD 01 10 29 7F AA AD 01
0368–10 29 80 CD 58 03 8D 58
0370–03 F0 03 8D 30 C0 A0 07
0378–88 D0 FD CA 30 02 D0 F6
0380–EE 61 03 EE 67 03 D0 D8
0388–EE 62 03 EE 68 03 AD 68
0390–03 C9 96 D0 CB A9 01 8D
0398–61 03 8D 67 03 A9 10 8D
03A0–62 03 8D 68 03 60 00 00
```

(2)将所需发声的内容先用录音机录到一段磁带上并将磁带倒回到内容开始位置。

(3)用转录电缆连接录音机喇叭输出端与 CEC–Ⅰ中华学习机的录音输入端。

(4)在监控状态下键入″300G″。然后按下录音机″PLAY″键，同时按下计算机的″RETURN″键。直到录制内容放完或荧光屏上光标重新出现，则按下录音机″STOP″键。

(5)在监控状态下键入″360G″。计算机喇叭中将放出预先录制的声音。

(6)录音机放音时音量放得小一些，则以后计算机放音的音质将要好一些。

(7)改变程序中 ＄31F 中(原值为 ＄05)和 ＄377 中(原值为 ＄07)的内容可改变放音的质量和放音时间的长短。规律是其值越小，音质越好，但时间越短。

(8)若希望将放音内容存到磁盘上，可用 DOS 命令“BSAVE 文件名，A＄1000,L＄7FFF″。

360. 怎样提高中华学习机读磁带的成功率

购置 CEC-I 型中华学习机而没有配置磁盘驱动器的情况是非常广泛的。从录音机磁带上读入游戏程序的工作对录音机质量要求很高。用普通的袖珍式立体声录放机或单放机无法将磁带上的内容读到中华学习机中。但是，如果按图 360-1 所示电路自制一个读带附加装置，则可在使用袖珍式录音机时将磁带上的内容无误地读入计算机；而且无论何种录音机，对音量旋钮旋置的位置几乎无关。

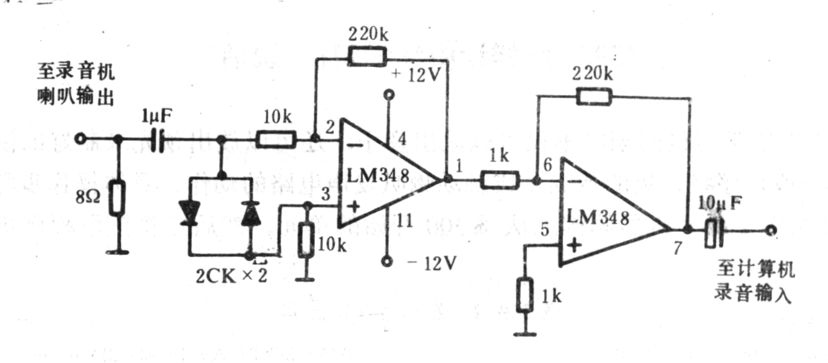
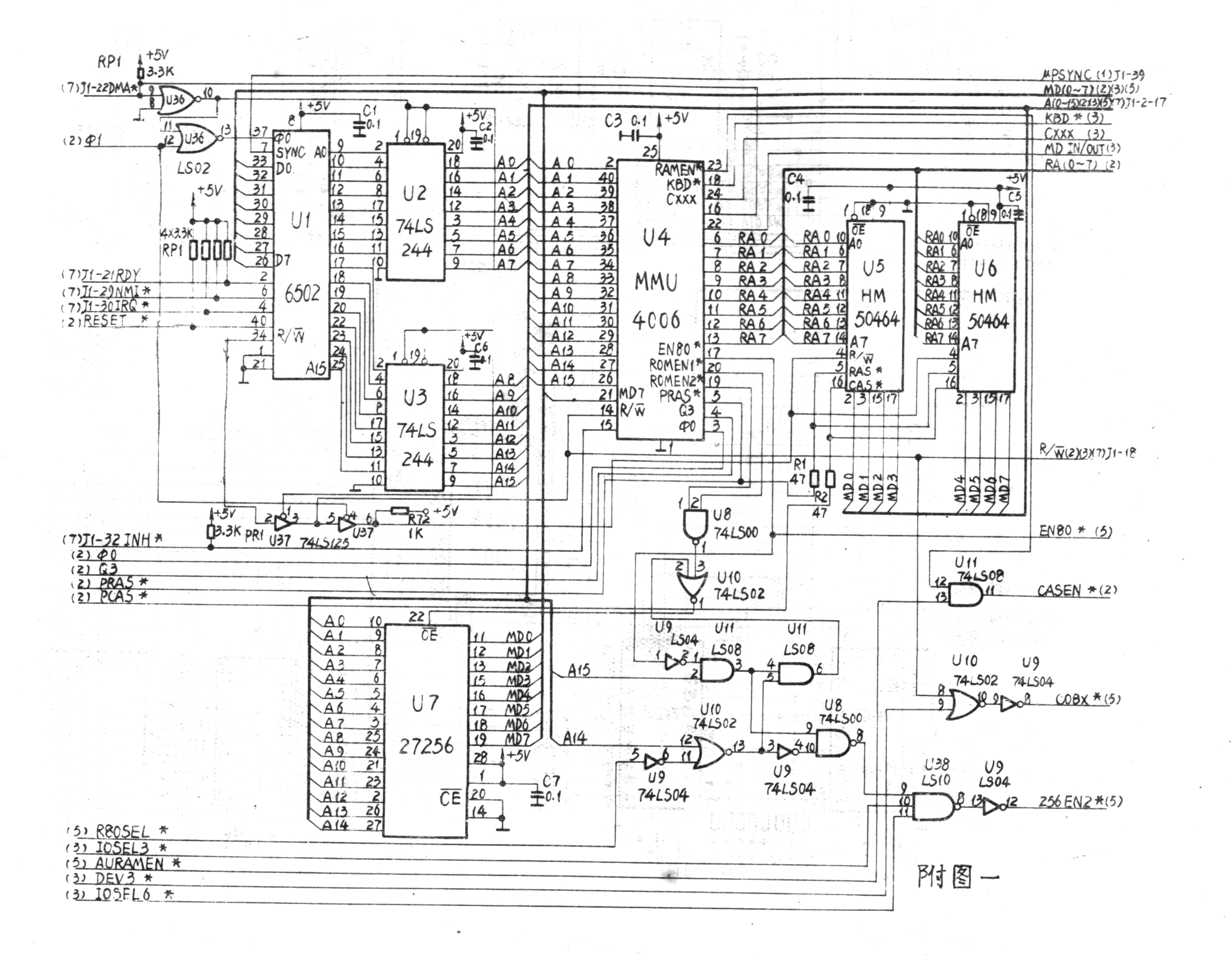
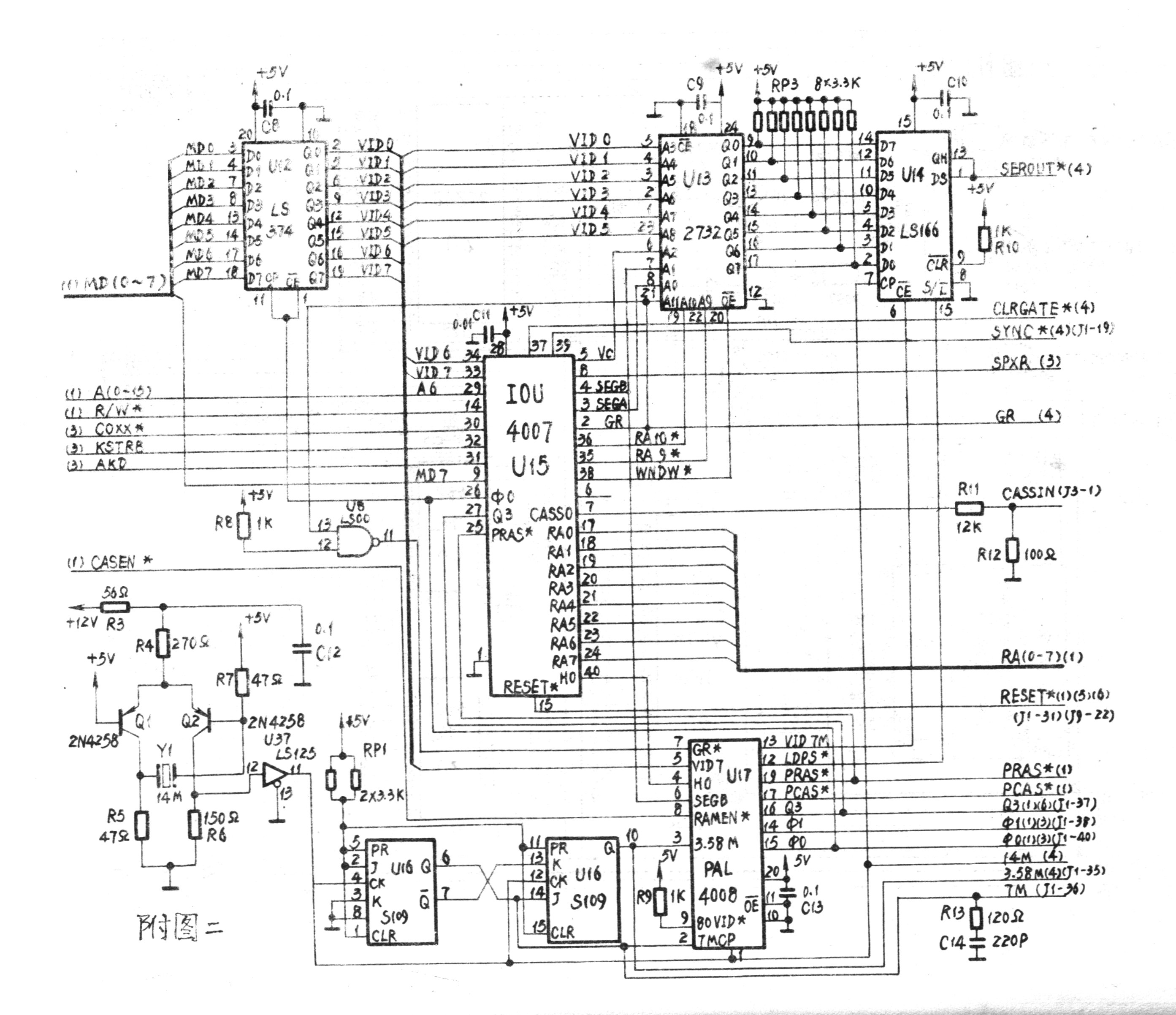


图 360-1　读带附加器电路原理图

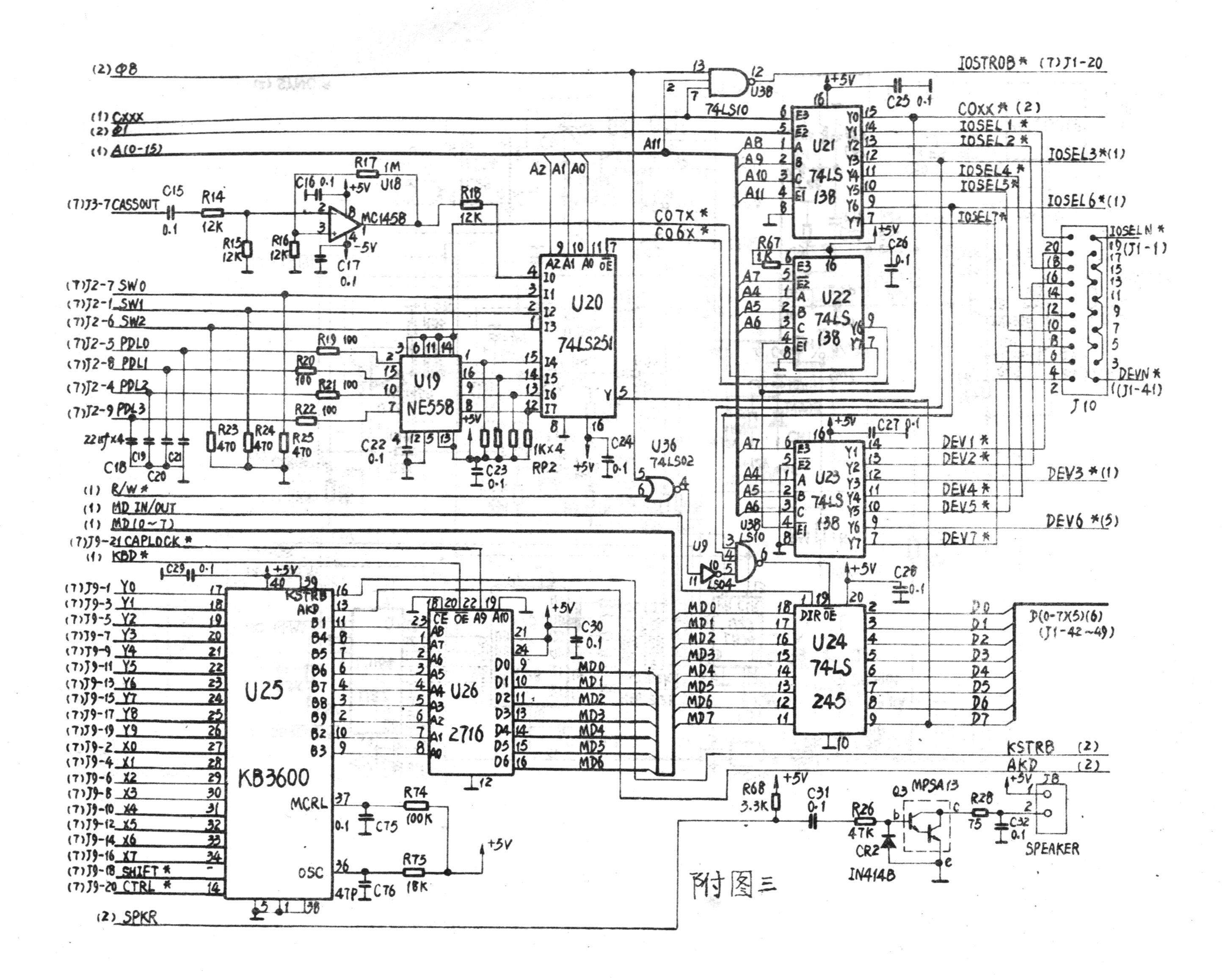
附加器的+12V 和-12V 电源可以从磁盘驱动器插座的第 13.15.17.19 脚(+12V)和第 9 脚(-12V)上取得。地线在第 1.3.5.7 脚上。


RP1
3.3K
(7)J1-22DMA*
(2) Φ1
U36
LS02
U1
6502
SYNC
R/W
4X3.3K
RP1
(7)J1-21RDY
(7)J1-29NMI *
(7)J1-30IRQ *
(2) RESET *
U2
74LS
244
U3
74LS
244
U4
MMU
4006
RAMEN*
KBD*
CXXX
EN80*
ROMEN1*
ROMEN2*
PRAS*
Q3
Φ0
MD7
R/W
U5
HM
50464
U6
HM
50464
RAS *
CAS *
MPSYNC (1)J1-39
MD(0~7)(2)(3)(5)
A(0~15)(2)(3)(5)(7)J1-2-17
KBD * (3)
CXXX (3)
MD IN/OUT (3)
RA(0~7) (2)
R/W(2)(3)(7)J1-18
EN80 * (5)
R1 47
R2 47
U8 74LS00
U10 74LS02
U37 74LS125
PR1 3.3K
R72 1K
(7)J1-32 INH *
(2) Φ0
(2) Q3
(2) PRAS *
(2) PCAS *
U7
27256
CE
A15
A14
U9 74LS04
U11 74LS08
CASEN * (2)
C0BX * (5)
U38 LS10
256EN2 * (5)
(5) R80SEL *
(3) IOSEL3 *
(5) AURAMEN *
(3) DEV3 *
(3) IOSEL6 *
附图一

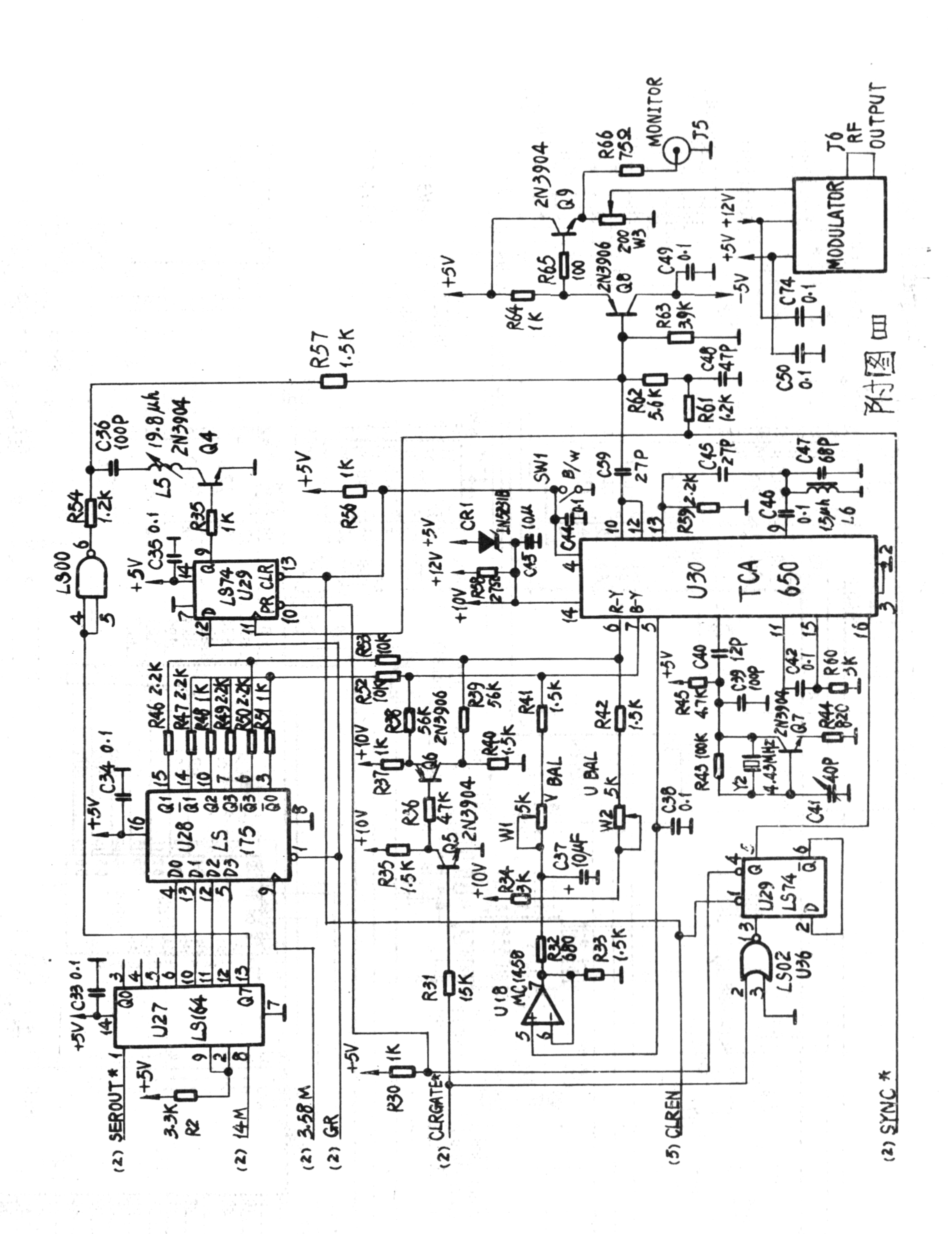


IOU
4007
U15
2732
U13
LS166
U14
LS 374
U12
PAL
4008
U17
S109
U16
LS125
U37
LS00
U8
2N4258
SEROUT*(4)
CLRGATE*(4)
SYNC*(4)(J1-19)
SPXR (3)
GR (4)
CASSIN (J3-1)
RA(0-7)(1)
RESET*(1)(5)(6)
(J1-31)(J9-22)
PRAS*(1)
PCAS*(1)
14M (4)
TM (J1-36)
MD(0~7)
A(0-15)
R/W*
COXX*
KSTRB
AKD
CASEN*
VID0
VID1
VID2
VID3
VID4
VID5
VID6
VID7
+5V
+12V
RP3 8×3.3K
RP1 2×3.3K
R3 56Ω
R4 270Ω
R5 47Ω
R6 150Ω
R7 47Ω
R8 1K
R9 1K
R10 1K
R11 12K
R12 100Ω
R13 120Ω
C14 220P
Y1 14M

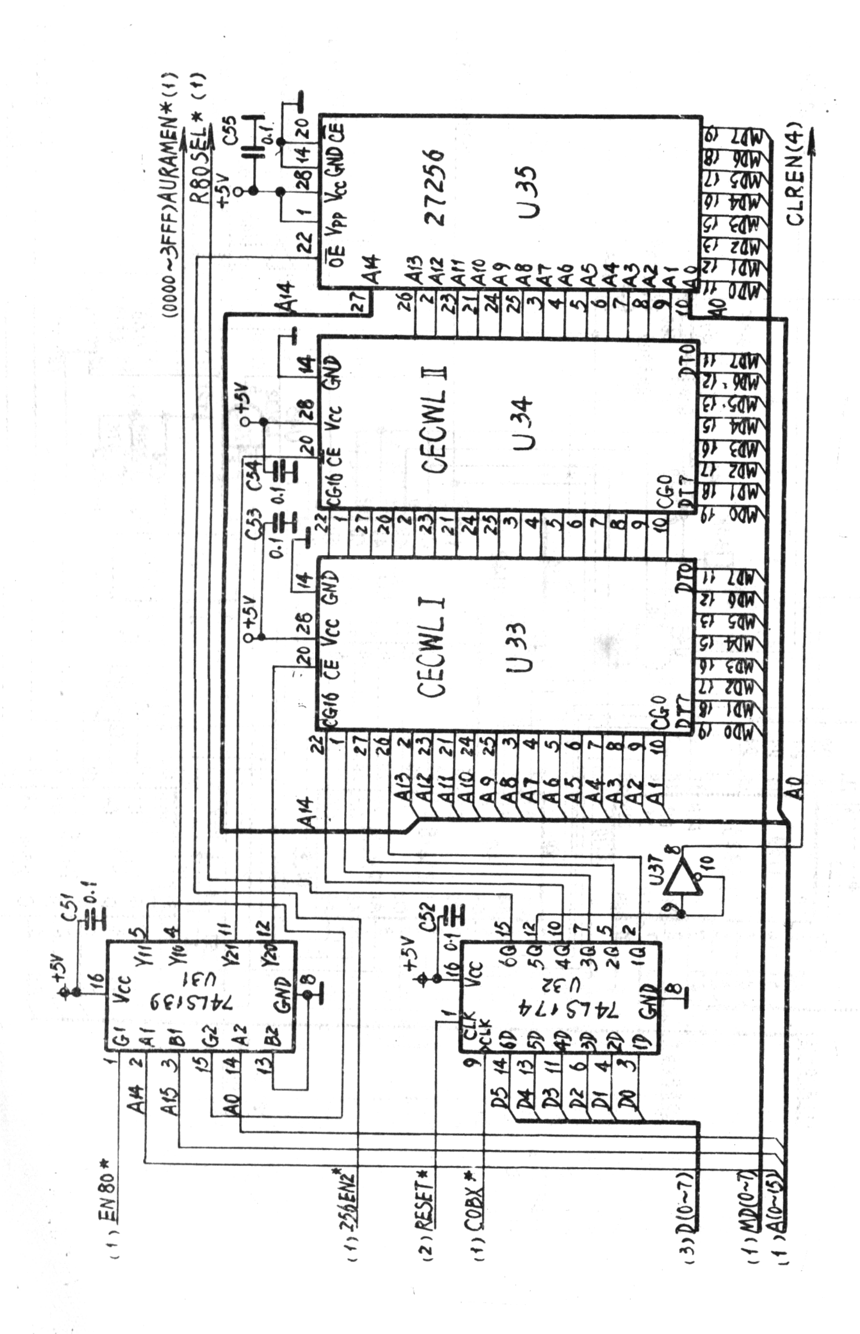
附图二


U21 74LS138
U22 74LS138
U23 74LS138
U24 74LS245
U20 74LS251
U19 NE558
U26 2716
U25 KB3600
U18 MC1458
Q3 MPSA13
CR2 IN414B
SPEAKER
J10
IOSTROB* (7)J1-20
COXX* (2)
D(0-7)(5)(6) (J1-42~49)
KSTRB (2)
AKD (2)

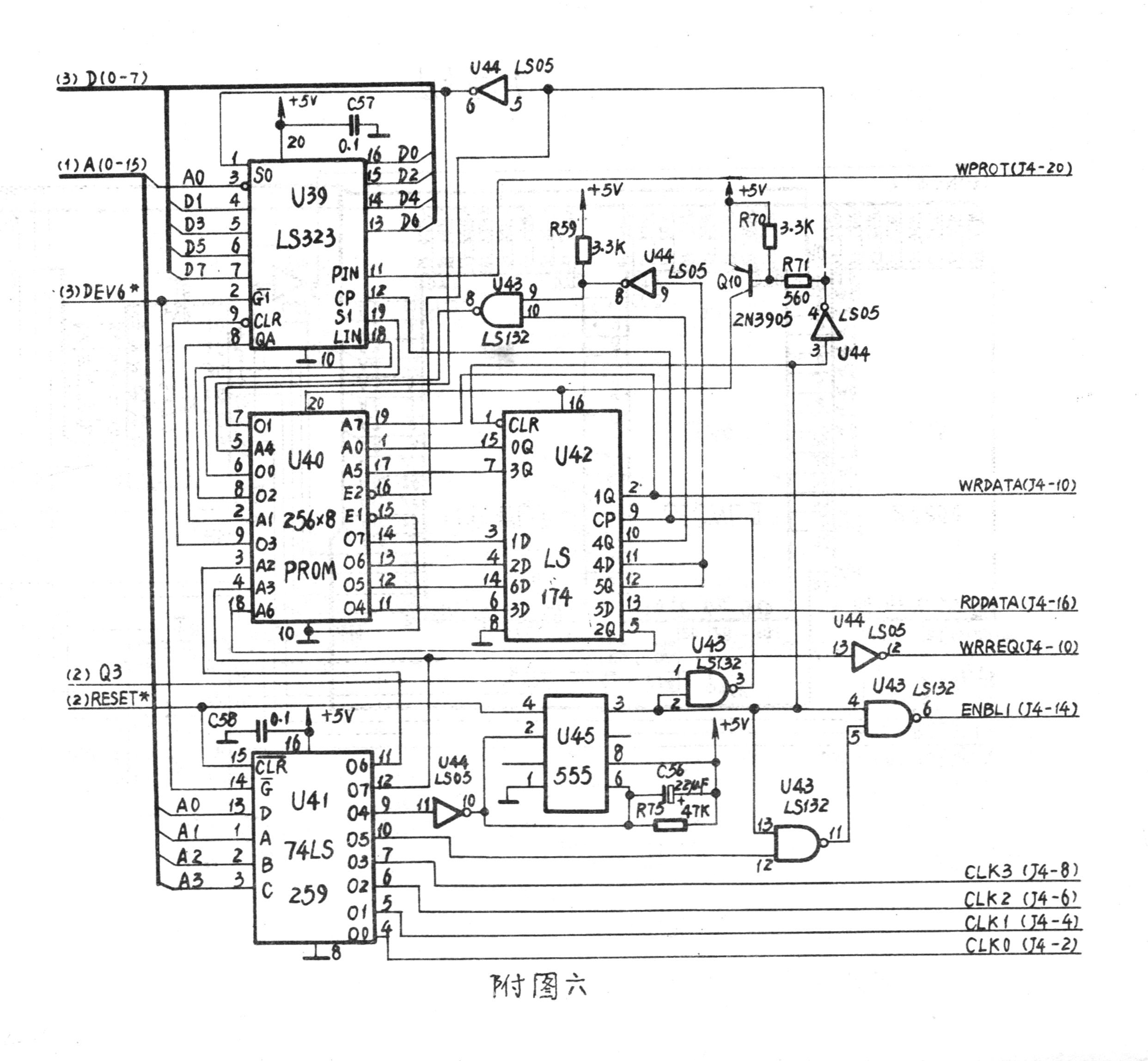
附图三



附图 III

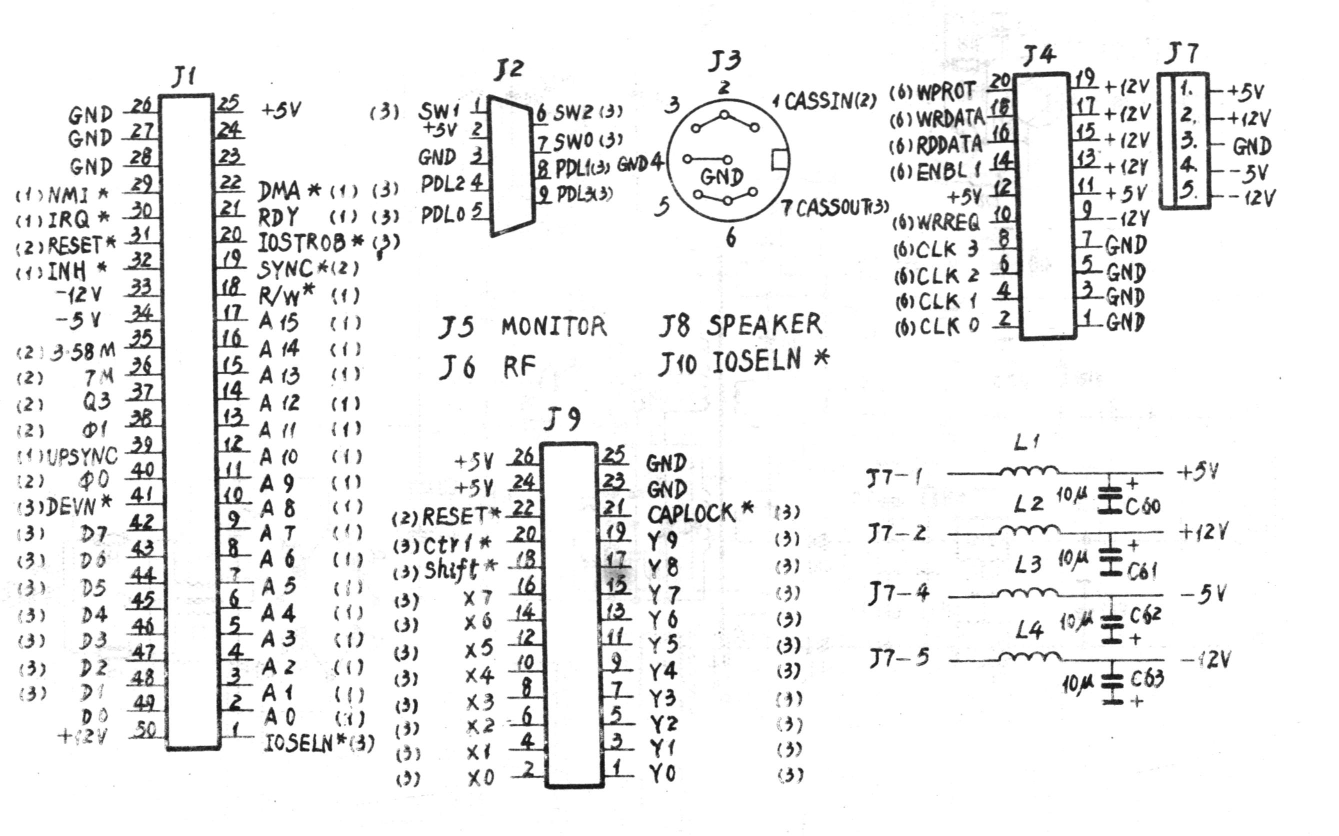

74LS139
U31
74LS174
U32
U37
CECWL I
U33
CECWL II
U34
27256
U35
C51
C52
C53
C54
C55
(0000~3FFF)AURAMEN*(1)
R80SEL*(1)
CLREN(4)
(1) EN80*
(1) 256EN2*
(2) RESET*
(1) COBX*
(3) D(0~7)
(1) MD(0~7)
(1) A(0~15)

附图 II

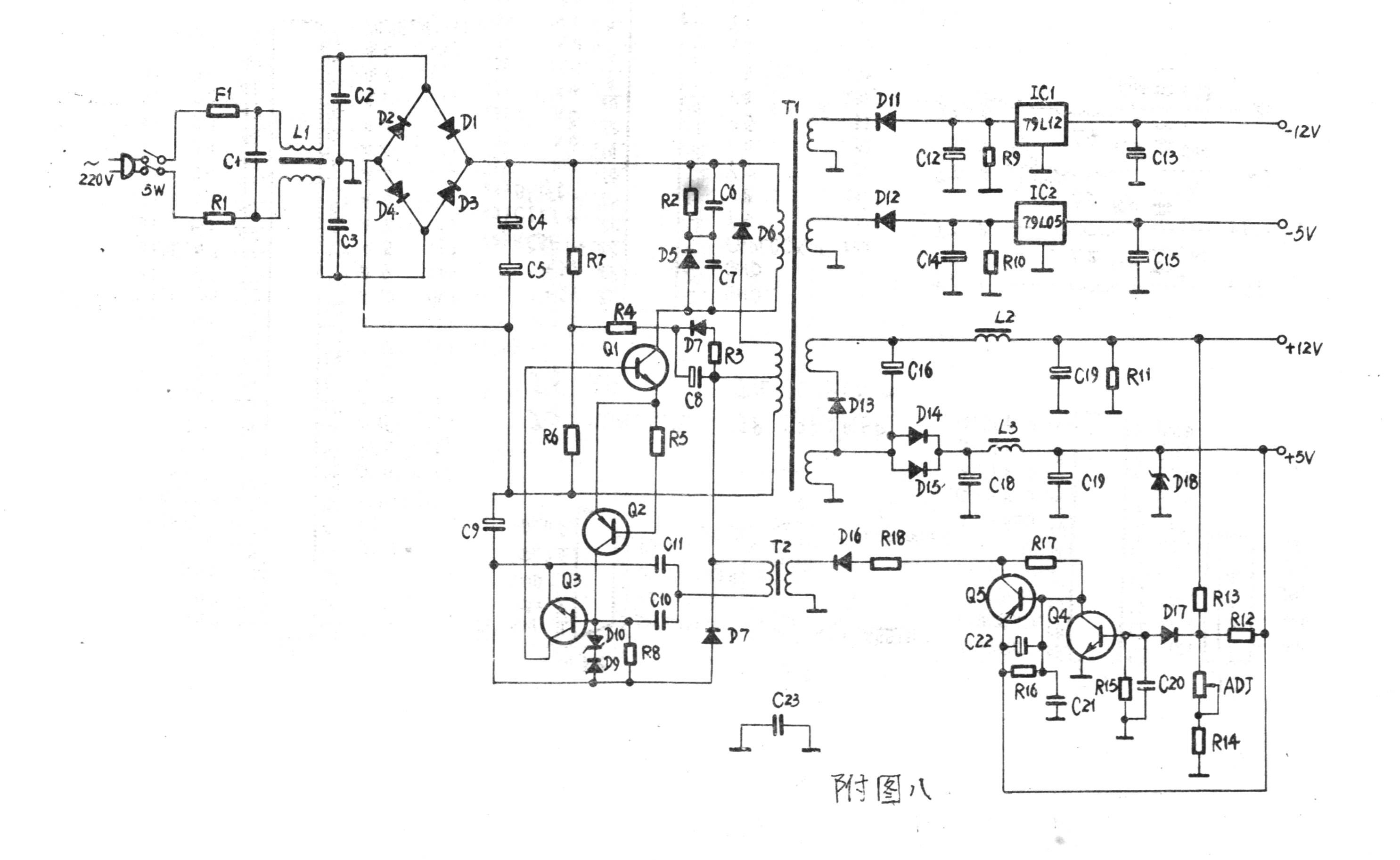


(3) D(0-7)
(1) A(0-15)
(3)DEV6*
(2) Q3
(2)RESET*
U39
LS323
U40
256×8
PROM
U42
LS
174
U41
74LS
259
U45
555
U44 LS05
U43 LS132
R59 3.3K
R70 3.3K
R71 560
Q10 2N3905
C57 0.1
C58 0.1
C56 22μF
R75 47K
+5V
WPROT(J4-20)
WRDATA(J4-10)
RDDATA(J4-16)
WRREQ(J4-10)
ENBL1 (J4-14)
CLK3 (J4-8)
CLK2 (J4-6)
CLK1 (J4-4)
CLK0 (J4-2)

附图六

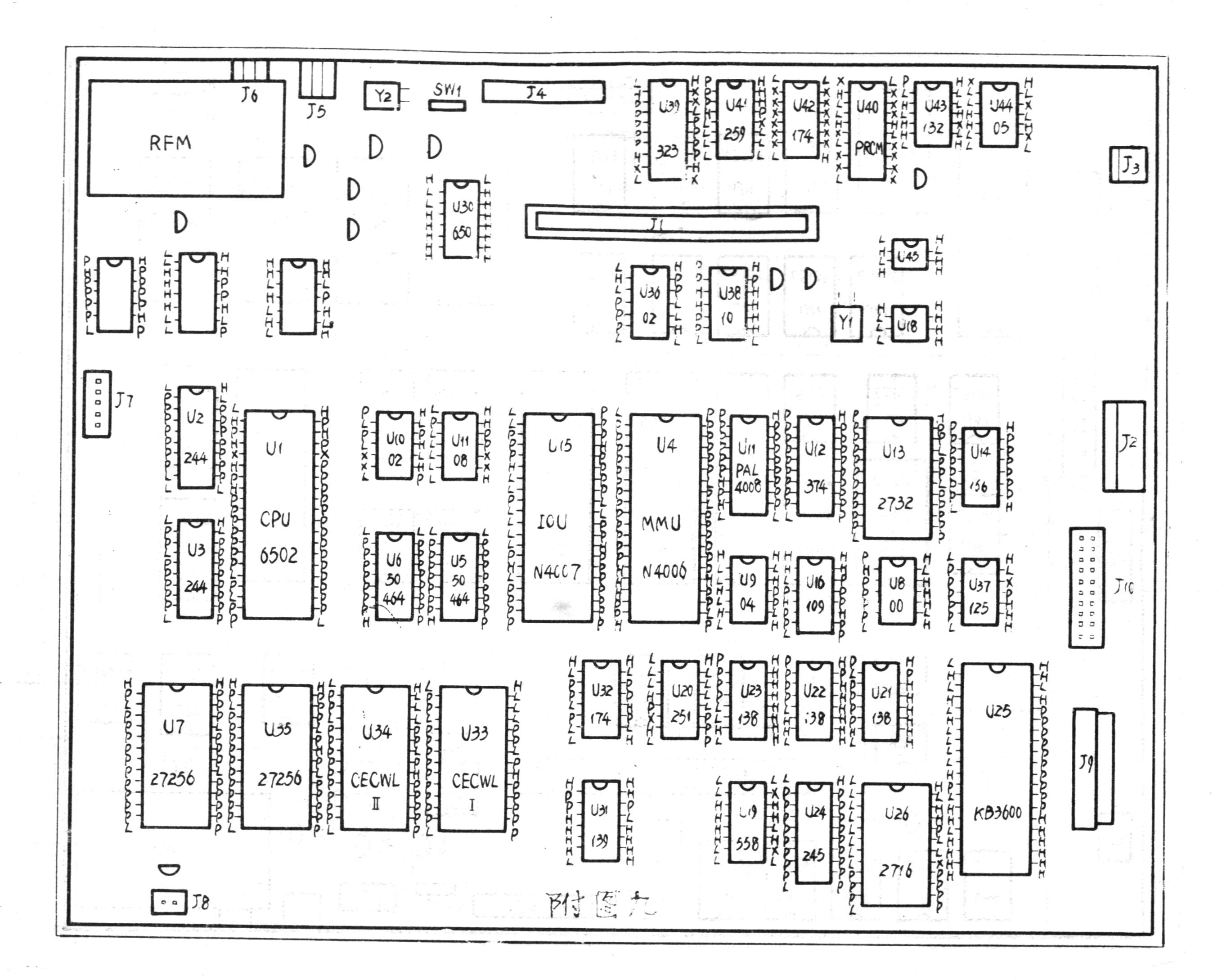

J1
J2
J3
J4
J7
J5 MONITOR
J6 RF
J8 SPEAKER
J10 IOSELN *
J9
CASSIN(2)
CASSOUT(3)
GND
L1
L2
L3
L4
C60
C61
C62
C63
10μ
J7-1
J7-2
J7-4
J7-5
+5V
+12V
-5V
-12V

附图七

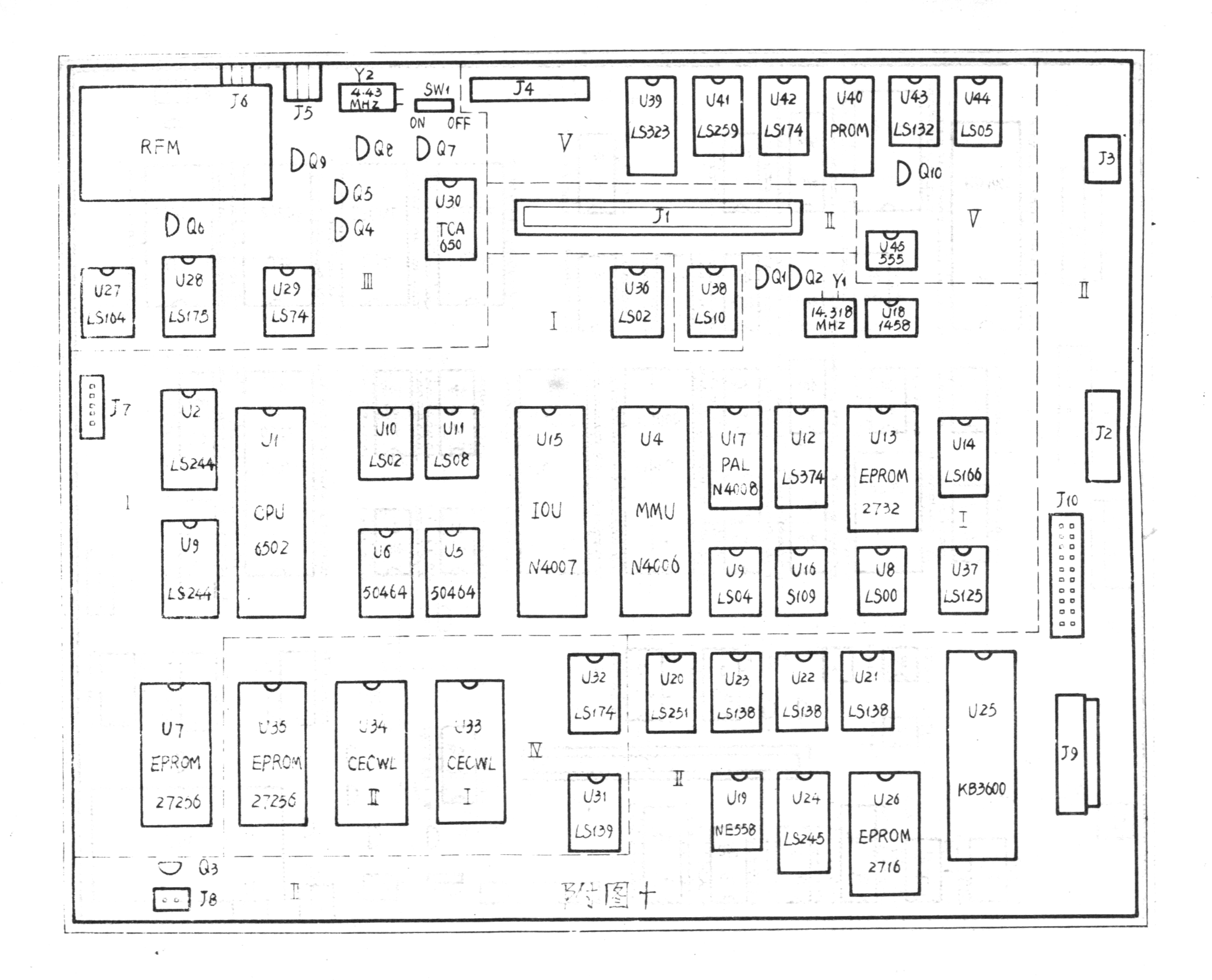


~220V
SW
F1
R1
C1
L1
C2
C3
D1
D2
D3
D4
C4
C5
R7
R4
Q1
R6
R5
R2
C6
D5
C7
D6
D7
R3
C8
Q2
C9
C11
Q3
C10
D10
D9
R8
D7
T1
T2
C23
D11
IC1
79L12
C12
R9
C13
-12V
D12
IC2
79L05
C14
R10
C15
-5V
L2
C16
C19
R11
+12V
D13
D14
L3
D15
C18
C19
D18
+5V
D16
R18
R17
Q5
Q4
C22
R16
C21
R15
C20
D17
R13
R12
ADJ
R14

附图八

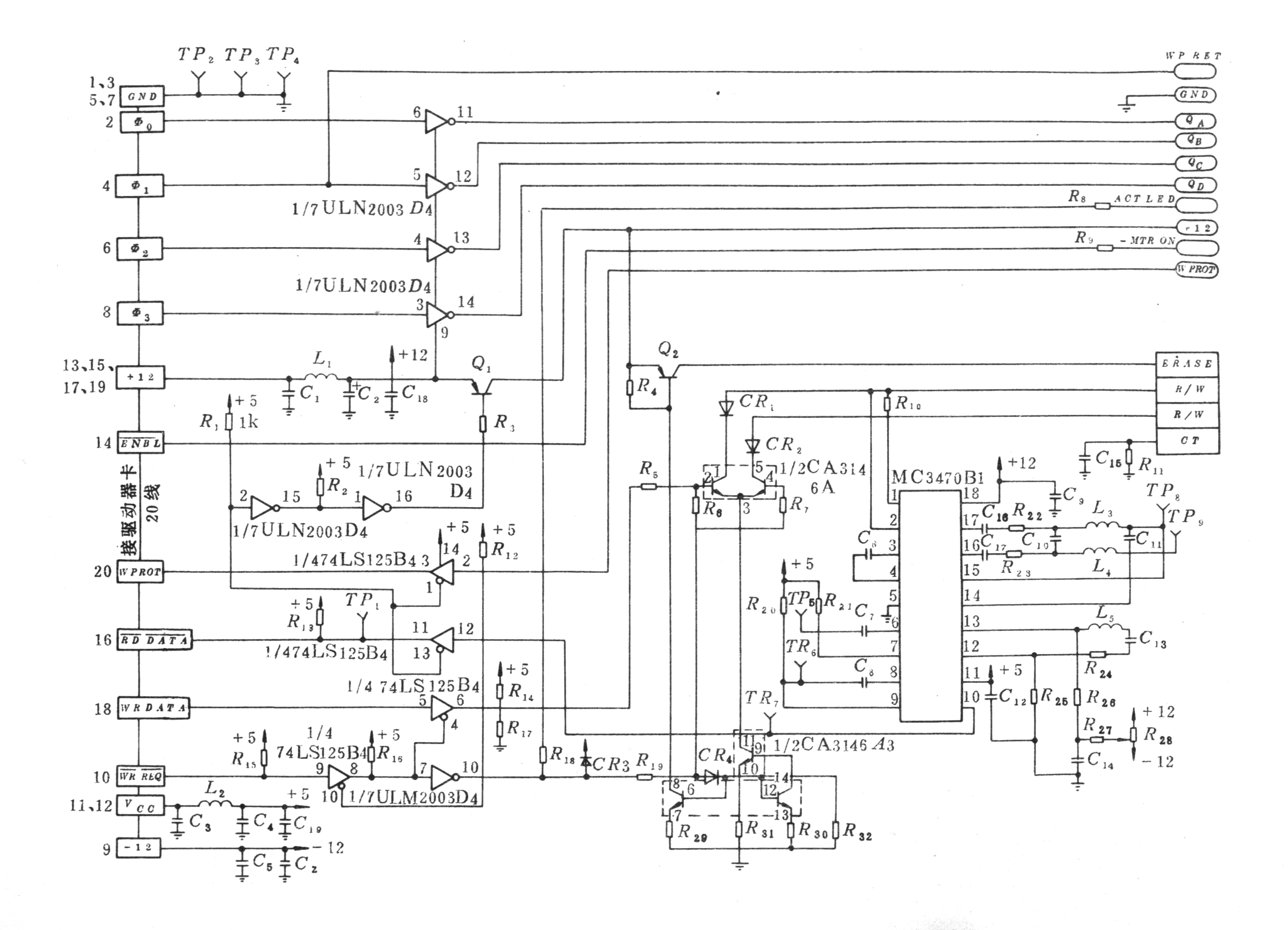

RFM
J6
J5
Y2
SW1
J4
U39 323
U41 259
U42 174
U40 PROM
U43 132
U44 05
J3
U30 650
J1
U45
U36 02
U38 10
Y1
U18
J7
U2 244
U1 CPU 6502
U3 244
U10 02
U11 08
U6 50 464
U5 50 464
U15 IOU N4CC7
U4 MMU N4006
U11 PAL 4008
U12 374
U13 2732
U14 156
J2
U9 04
U16 109
U8 00
U37 125
J10
U7 27256
U35 27256
U34 CECWL II
U33 CECWL I
U32 174
U20 251
U23 138
U22 138
U21 138
U25 KB3600
J9
U31 139
U19 558
U24 245
U26 2716
J8

附图九



RFM
J6
J5
Y2 4.43 MHz
SW1 ON OFF
J4
U39 LS323
U41 LS259
U42 LS174
U40 PROM
U43 LS132
U44 LS05
J3
Q9 Q8 Q7 Q5 Q4 Q6 Q10
U30 TCA 650
J1
U45 555
U27 LS164
U28 LS175
U29 LS74
U36 LS02
U38 LS10
Q1 Q2 Y1
14.318 MHz
U18 1458
J7
U2 LS244
U1 CPU 6502
U10 LS02
U11 LS08
U15 IOU N4007
U4 MMU N4006
U17 PAL N4008
U12 LS374
U13 EPROM 2732
U14 LS166
J2
J10
U9 LS244
U6 50464
U5 50464
U9 LS04
U16 S109
U8 LS00
U37 LS125
U7 EPROM 27256
U35 EPROM 27256
U34 CECWL
U33 CECWL
U32 LS174
U20 LS251
U23 LS138
U22 LS138
U21 LS138
U25 KB3600
J9
U31 LS139
U19 NE558
U24 LS245
U26 EPROM 2716
Q3
J8

附图十


TP2 TP3 TP4
GND
Φ0
Φ1
Φ2
Φ3
+12
ENBL
WPROT
RD DATA
WR DATA
WR REQ
VCC
-12
接驱动器卡
20线
1/7ULN2003 D4
1/7ULN2003D4
1/474LS125B4
1/4 74LS125B4
1/7ULM2003D4
1/2CA3146A
1/2CA3146 A3
MC3470B1
WP RET
QA
QB
QC
QD
ACT LED
-MTR ON
ERASE
R/W
CT


中华学习机使用技巧与检修360问
電子工業出版社

● 责任编辑：徐云鹏
● 封面设计：阎欢玲



● 为满足广大中华学习机爱好者的需要、电子工业出版社《电子与电脑》杂志编辑部向您推荐下列学习机功能扩展卡：

● 1. 中华学习机、APPLE Ⅱ微机Z80/PRT卡

● 2. AP51-Ⅰ型单片微机学习、开发卡（万能接口卡）

● Z80/PRT卡是专为CEC-Ⅰ型中华学习机、APPLE Ⅱ设计的Z80卡。插上该卡后，可使用CP/M操作系统，运行CP/M操作系统下的COBOL程序、FORTRAN程序和dBASE Ⅱ数据库管理程序。Z80/PRT卡上不仅有Z80CPU芯片及其外围电路，同时还做上了打印机接口电路，具有驱动打印机 机的功能。

● AP51-Ⅰ型单片微机学习、开发卡（万能接口卡）是一种用于学习和开发MCS-51系列单片微机的多功能卡，当插在中华学习机或APPLE Ⅱ机上时可以做为一个串行通讯接口卡（用于中华学习机、APPLE Ⅱ微机与IBMPC等微机之间通讯、联网）。或做为一个打印机接口卡（可用于全脱机方式打印），或做为一个时钟卡（可显示时间、控制主机定时动作），或做为并行接口卡，或做为PC机磁盘驱动器兼容卡……。只要配上相应的软件就可以方便地用于再开发，灵活地完成各种功能。

ISBN 7—5053—0661—8 / TP·108　定价：11.00 元